



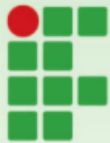
INSTITUTO FEDERAL
Santa Catarina



Processo de Desenvolvimento

Edjandir Corrêa Costa

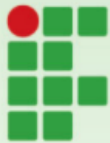
edjandir.costa@ifsc.edu.br



Processo de Desenvolvimento

- **Definição:**

- É um roteiro que determina quais são as tarefas necessárias e em que ordem elas devem ser executadas para construir softwares de qualidade;
- Ele organiza uma atividade que pode, sem controle, tornar-se caótica;
- Há uma imensa diversidade de processos de software
 - ✓ Não existe o ideal;
- Tradicionalmente, é adaptado ao tipo de software que se está construindo.

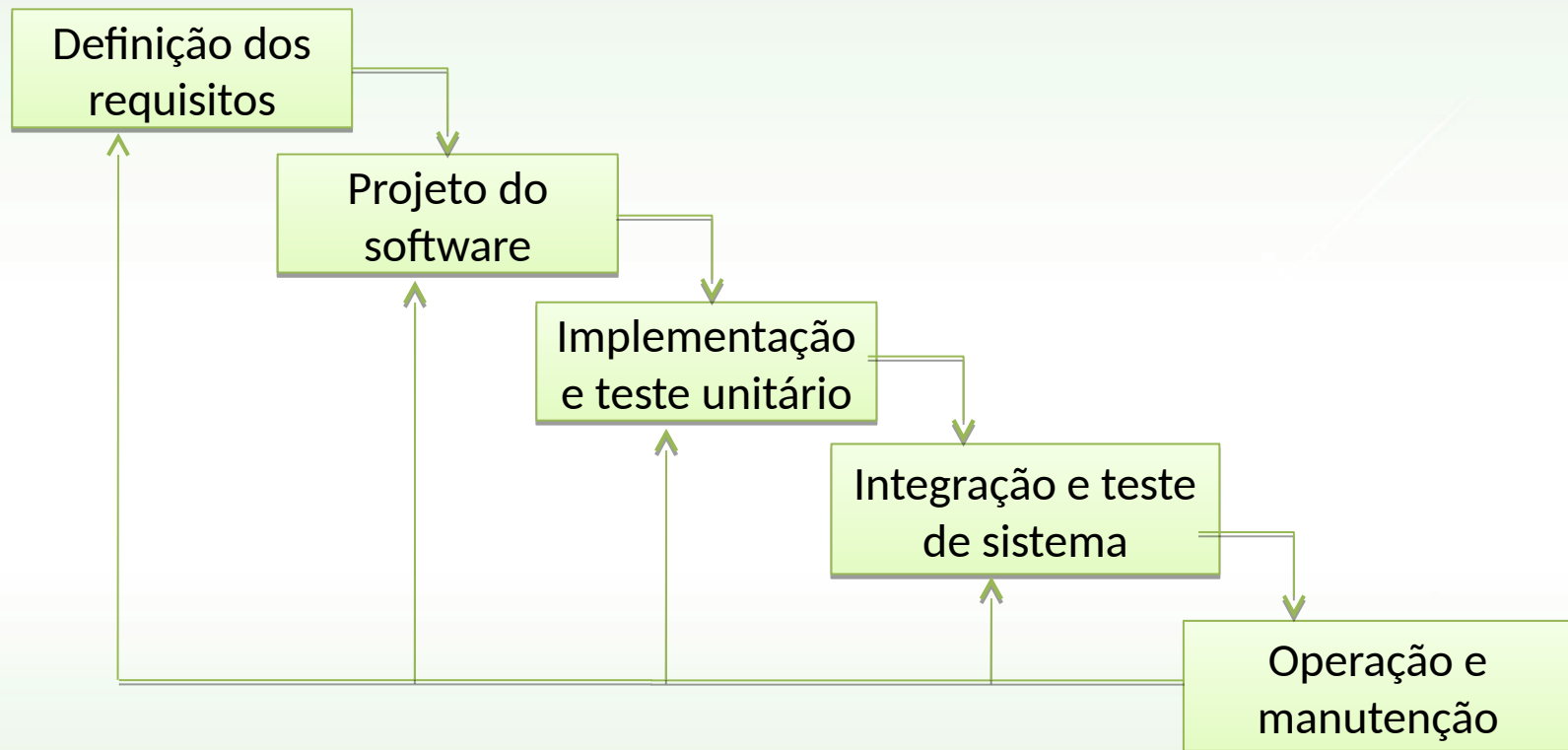


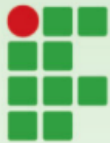
Modelo em Cascata

- Proposto em 1970, também chamado de:
 - *Waterfall* (queda d'água), linear ou clássico
- É caracterizado por uma sequência que se estabelece no desenvolvimento de um software, onde:
 - Uma fase ocorre sempre após o término de uma outra fase;
 - Cada fase gera um conjunto de documentos que são usados pela fase seguinte;
 - Seu principal ponto forte está na simplicidade, pois as fases estão bem definidas e organizadas.

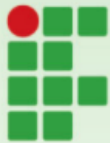


- Ciclo de vida (fases ou etapas)

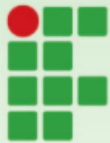




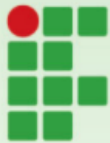
- **Funcionamento básico:**
 - Definição de requisitos:
 - ✓ Objetivos e restrições são obtidas via consulta a usuários;
 - ✓ São definidos e completamente detalhados.
 - Projeto de software:
 - ✓ Estabelece a solução tecnológica para o conjunto de requisitos obtidos.
 - Implementação e teste unitário:
 - ✓ Envolve a codificação do projeto de software em alguma linguagem de programação;
 - ✓ Faz-se verificações de cada módulo com objetivo de observar se eles atendem a especificação.



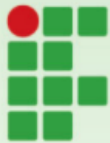
- **Funcionamento básico (cont.):**
 - Integração e teste de sistema:
 - ✓ Os módulos (programas) são integrados e testados como um sistema completo;
 - ✓ Software é liberado para o cliente avaliar.
 - Manutenção e operação:
 - ✓ Software é instalado e colocado em operação.



- **Problemas/Limitações:**
 - O cliente deve ser paciente, pois não existe previsão de entrega de versões funcionais do software no decorrer do desenvolvimento:
 - ✓ Ele só terá contato com o software quando ele estiver concluído.



- **Problemas/Limitações (cont.):**
 - Nem sempre é possível descobrir tudo aquilo que o cliente deseja, ou seja, nem todos os requisitos são descobertos por inteiro:
 - ✓ Em muitas situações estes requisitos são identificados em fases finais do modelo o que pode resultar em retrabalhos e maiores custos de desenvolvimento já que a descoberta tardia de um requisito pode modificar a arquitetura ou estrutura do software.



- **Problemas/Limitações (cont.):**
 - Não há interatividade (participação) com o cliente, pois ele deve esperar a execução de todas as fases para ter o software;
 - Se ocorrer um atraso em alguma fase, todo o processo é afetado;
 - Não suporta modificações nos requisitos durante o desenvolvimento.



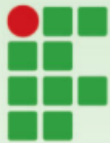
- **Uso se restringe quando:**
 - Os requisitos forem inicialmente bem compreendidos e conhecidos por completo;
 - Houver pouca probabilidade de mudança de requisitos.



INSTITUTO FEDERAL
Santa Catarina

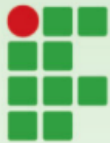


Modelos Evolucionários



Modelo de Prototipagem

- Sua concepção foi inspirada em dois problemas:
 - Dificuldade de identificar e compreender requisitos por completo
 - ✓ Desenvolvimento de protótipos para ajudar a esclarecer requisitos;
 - Falta da participação do cliente no desenvolvimento
 - ✓ Permitir ao cliente que faça experiências com o protótipo já no início do desenvolvimento, apontando deficiências e ajustes a serem implementados na próxima versão do protótipo (*feedback* rápido do cliente);



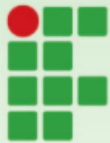
Modelo de Prototipagem

- Protótipo pode ser:
 - Interfaces (janelas);
 - Subconjunto de alguma funcionalidade.



- Ciclo de vida





- **Problemas/Limitações:**

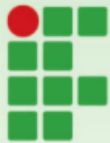
- **O processo não é visível:** como o sistema é desenvolvido rapidamente, não há tempo de documentar as versões;
- **Os sistemas podem ser mal-estruturados:** mudanças constantes podem corromper a estrutura do software.
- **Para sistemas grandes,** onde diversas equipes desenvolvem partes do sistema, é difícil estabelecer uma arquitetura estável do software usando esta abordagem;
- O cliente deve estar ciente de que o protótipo é usado apenas para levantar/ confirmar requisitos.



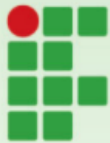
INSTITUTO FEDERAL
Santa Catarina



Modelos Iterativos



- São baseados em ciclos de atividades, ou seja, a especificação ocorre junto com o desenvolvimento do software.
 - Há dois tipos básicos:
 - ✓ Espiral
 - ✓ Incremental

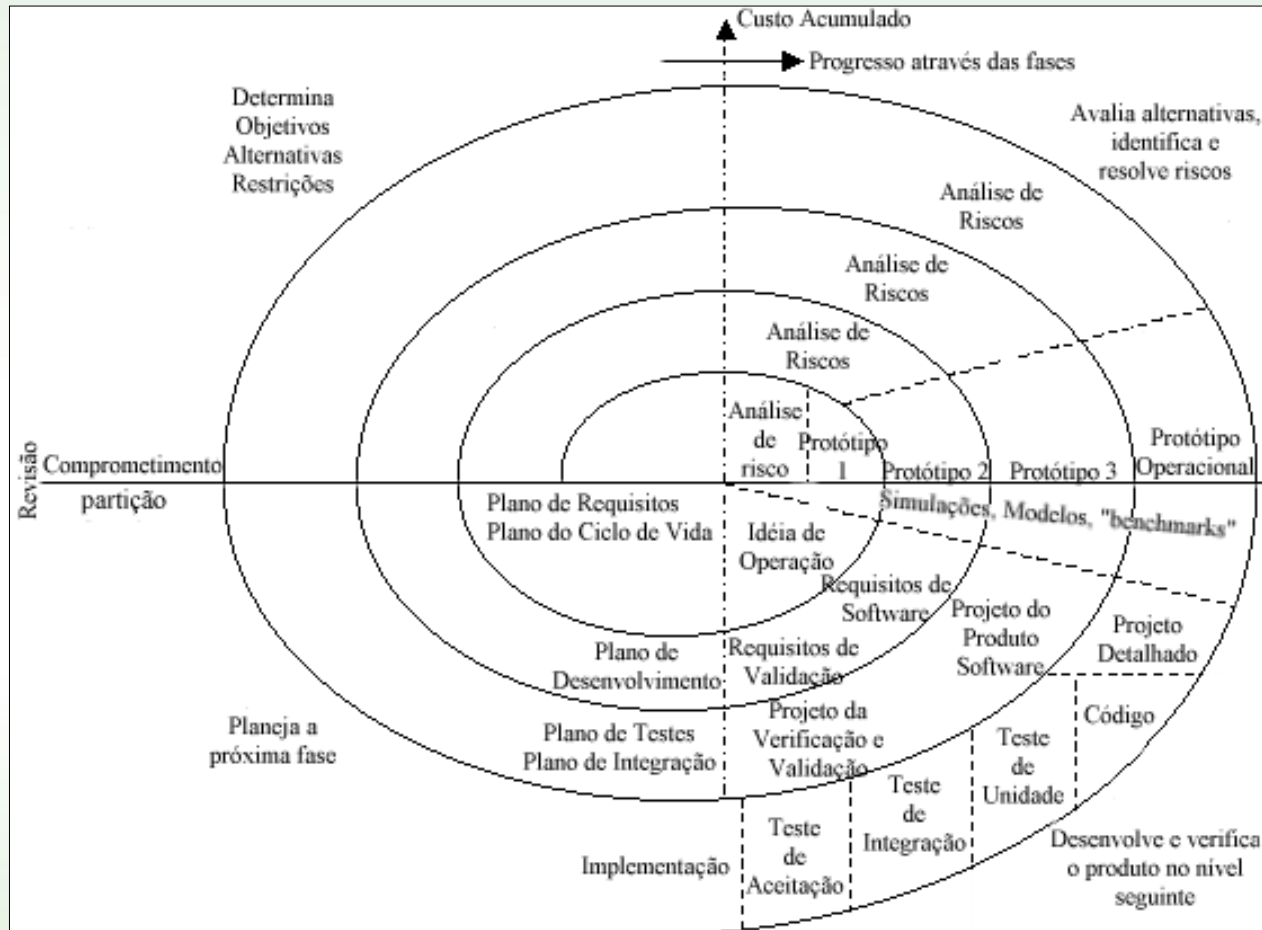


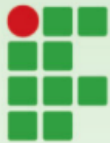
Modelo Espiral

- Em 1988, Boehm sugeriu o modelo espiral para o desenvolvimento de software:
 - É um modelo que abrange as melhores características dos modelos em cascata e prototipação acrescentando um novo elemento: a análise de riscos.



- Ciclo de vida





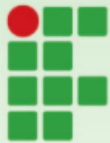
- **Funcionamento básico:**

- Um ciclo da espiral inicia com a elaboração dos objetivos;
- Estratégias são enumeradas e avaliadas em relação a cada objetivo e as fontes de riscos de projeto são identificadas;
- Resolvem-se esses riscos através da coleta de dados, prototipação e simulação;
- Parte-se, então, para o desenvolvimento, seguido do planejamento para a próxima fase.



- **Vantagem:**
 - Versões do sistema, obtidas a cada ciclo, cada vez mais completas.

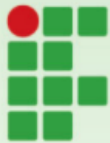
- **Problemas/Limitações:**
 - É um modelo relativamente novo, sem ter sido amplamente usado;
 - Exige-se experiência para avaliação dos riscos, pois caso algum risco não seja descoberto e resolvido poderá gerar problemas futuros.



Modelo Incremental

- **Características gerais:**

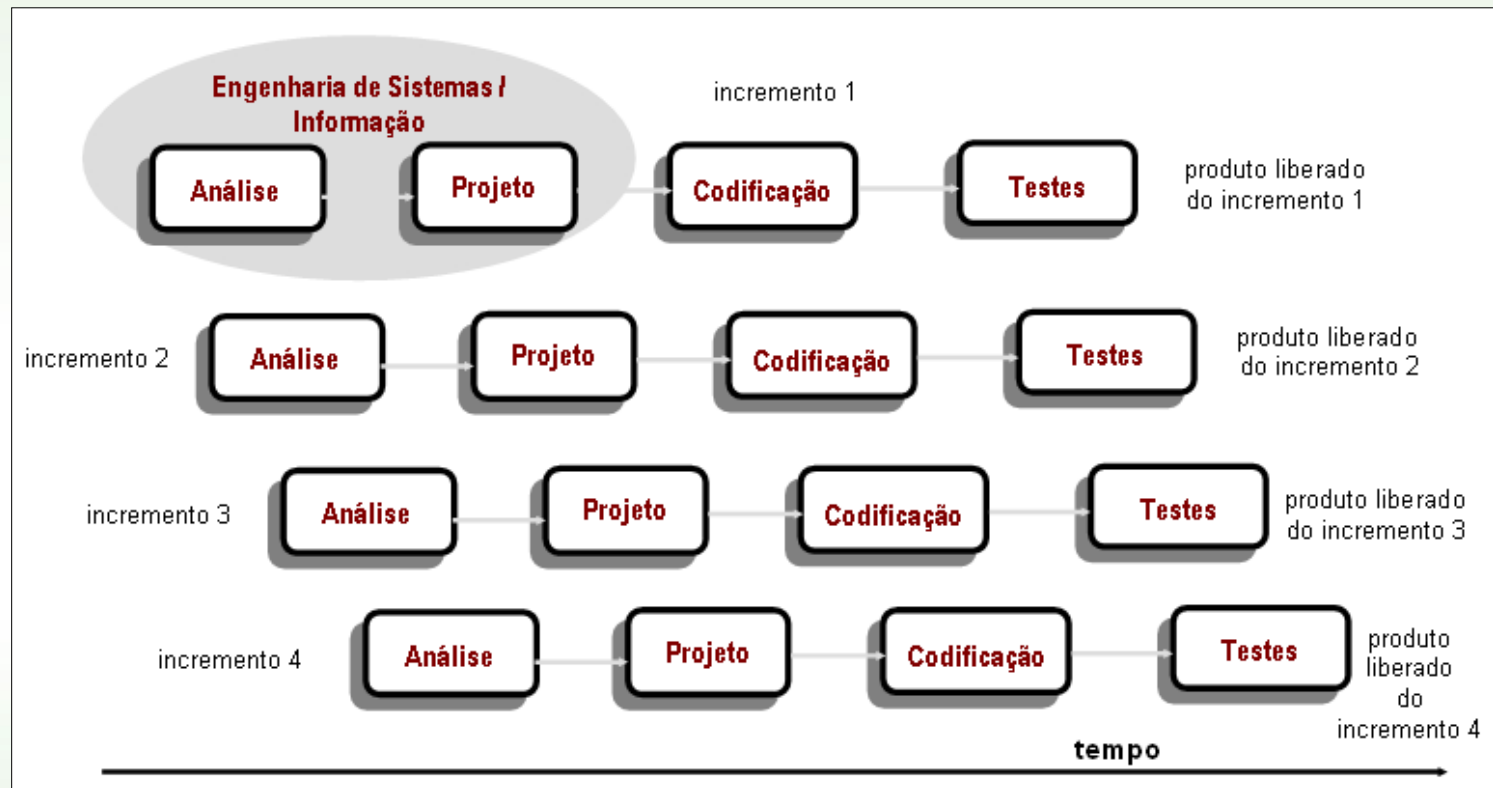
- As versões do software evoluem para o sistema final;
- Parte-se de um conjunto inicial importante para depois atacar os requisitos menos importantes;
- Incrementos devem ser pequenos com tempo definido;
- Incrementos devem envolver ao menos uma funcionalidade;

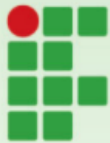


- **Características gerais (cont.):**
 - No incremento atual, os requisitos não podem ser mudados;
 - Cada sequência produz incrementos do software passíveis de serem entregues;
 - Há progressivamente mais funcionalidades a serem entregues;
 - O primeiro incremento é chamado de núcleo de produto.



— Ciclo de vida





Bibliografia

- PRESMAN, Roger. **Engenharia de Software**. 5a ed. McGraw-Hill: São Paulo, 2002.
- SOMMERVILLE, Ian. **Engenharia de Software**. 9a ed.: Pearson: Rio de Janeiro, 2011.