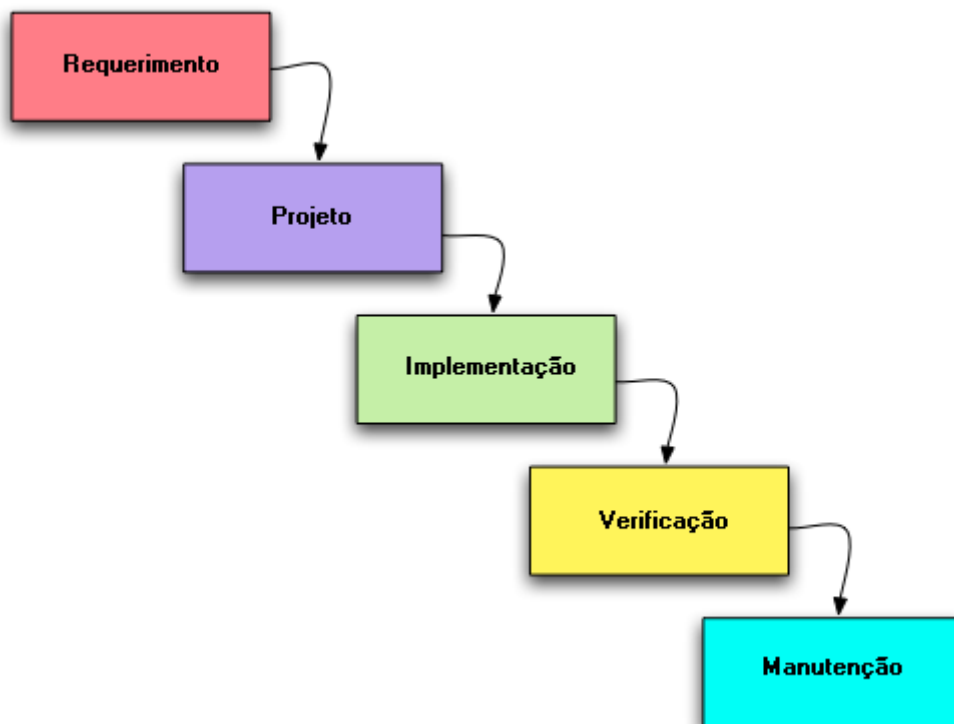


Engenharia de Software

Acadêmico: Cárliton Pedrosa Pessoa

Resumo Modelo de Cascata

O modelo de cascata é um modelo de desenvolvimento de software sequencial, passando para a próxima fase somente quando a fase anterior está completa e perfeita. Através das fases de análise de requisitos, projeto, implementação, teste(validação), integração e manutenção do software.



- **Requerimento:** Funções e restrições são definidas, com ajuda de clientes e usuários, indicando o que deve ser implementado.
- **Projeto:** Indica como o software deve ser implementado.
- **Implementação:** As unidades do software devem ser codificadas e testadas individualmente.
- **Verificação:** As unidades são testadas.
- **Manutenção:** O sistema é instalado e colocado em operação. A manutenção envolve a correção de erros e evolução do sistema para atender os novos requisitos.

O modelo cascata foi proposto em 1970 por W. W. Royce, sendo o único modelo com aceitação geral até 1980. A versão original vem sendo retocada e melhorada com o passar dos anos, o modelo cascata é um dos mais importantes e mais usados em projetos modernos.

O modelo cascata é apropriado quando se tem um entendimento claro dos requisitos mas também tem dificuldade em realizar mudanças com o processo em andamento.

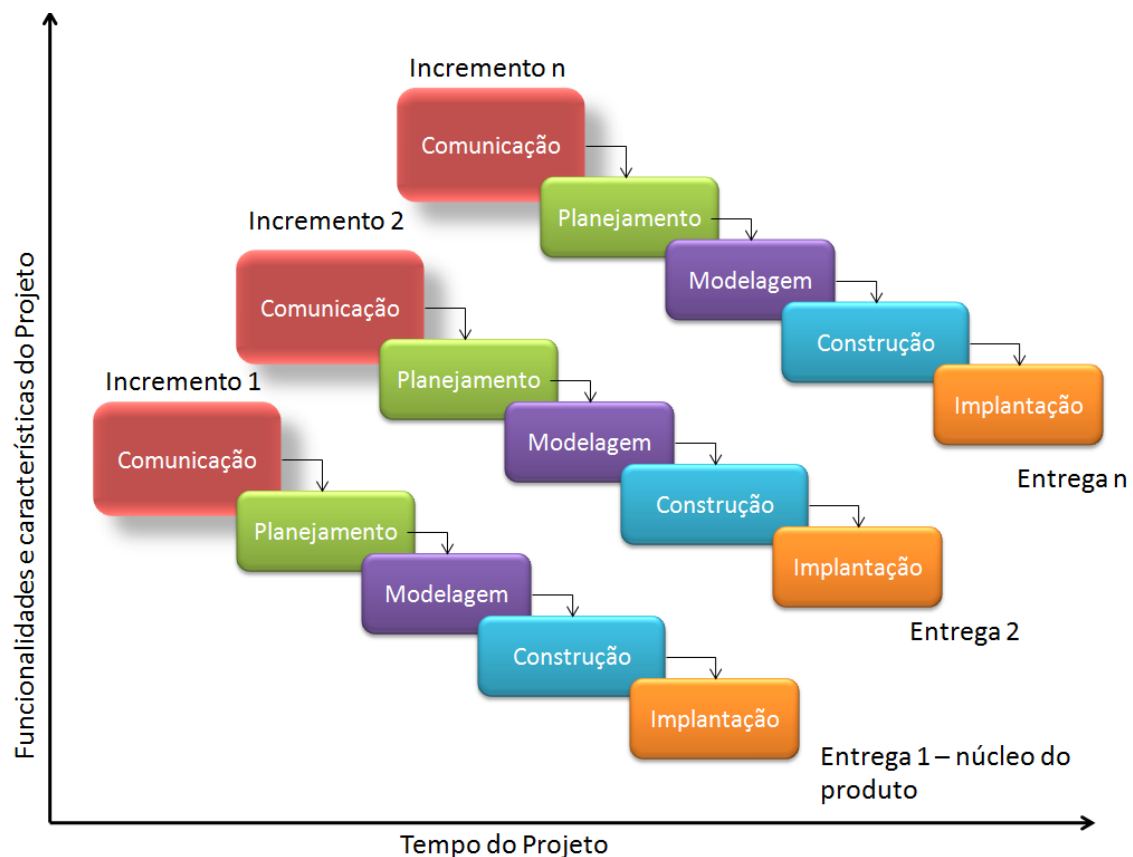
Definido o modelo de desenvolvimento, existem três tipos de abordagens para implementá-lo:

Cascata pura: o projeto progride de forma sequencial, partindo da especificação de requisitos até a manutenção, realizando uma revisão ao final de cada etapa para determinar se ela está pronta ou não para passar para a seguinte, permanecendo caso não esteja pronto.

É um modelo de baixo custo por ajudar a localizar erros nas primeiras etapas do projeto.

Incremental: é uma estratégia de planejamento estagiado em que várias partes do sistema é desenvolvido em paralelo, e integradas quando completadas. A alternativa ao desenvolvimento incremental é desenvolver todo o sistema com uma integração única.

Esse modelo precisa ser relativamente pequeno, é flexível e fácil de gerenciar processos administrativos, sendo possível fazer um software com uma estrutura melhor, assim os testes tornam se simples.



Evolucionário: tem como característica a possibilidade de desenvolver versões mais completas de um software, e quando houver a necessidade, ocorre mudanças e atualizações no produto.

Com a participação do usuário acontece o refinamento do software, com um feedback rápido.