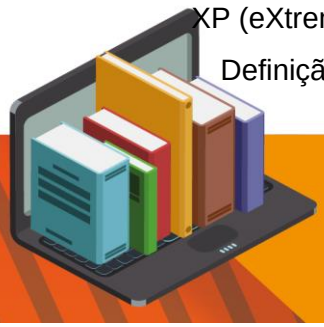
concursos de ti

ebooks e mentoria

ebook

engenharia de software

Sumário



Conceitos básicos

Definição e princípios

Primeiramente é interessante saber a definição de Engenharia de Software de acordo com a IEEE, com Friedrich Bauer e com os renomados Pressman e Sommerville:

- A **IEEE** define a Engenharia de Software como a aplicação de uma abordagem sistemática, disciplinada e quantificável de desenvolvimento, operação e manutenção de software.

- **Friedrich Bauer** conceitua como a criação e a utilização de sólidos princípios da engenharia a fim de obter software de maneira econômica que seja confiável e que rode em máquinas reais.

- **Pressman:** A Engenharia de Software ocorre como consequência da Engenharia de Sistemas. A engenharia de sistemas está preocupada com todos os aspectos do desenvolvimento de sistemas computacionais: engenharia de hardware, engenharia de software e engenharia de processos.

- **Sommerville:** A Engenharia de Software não está relacionada apenas com os processos técnicos de desenvolvimento de software, mas também com atividades de gerenciamento de projeto de software e desenvolvimento de ferramentas, métodos e teorias que apoiem a produção de software.

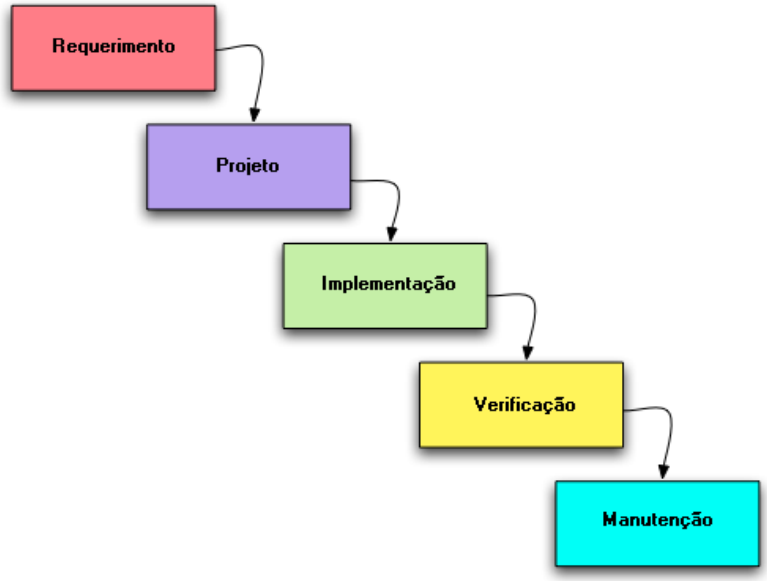
Princípios da Engenharia de Software:

- Formalidade
- Abstração
- Generalização
- Flexibilidade
- Decomposição

Principais Modelos de Desenvolvimento de Software

Modelo Cascata

Também conhecido como Sequencial, Linear, Cascata, Waterfall, Clássico, Rígido, Monolítico. As fases são bem definidas. Uma só começa após o fim da outra. Uma é requisito da outra. O cliente só vê um artefato no final do processo. Os defeitos só são descobertos na fase de teste. É recomendável que os requisitos já estejam bem definidos e não vão mudar, pois esse modelo não recebe bem mudanças nos requisitos. Funciona bem com projetos pequenos e simples em que os requisitos são bem conhecidos. Não funciona bem com requisitos complexos e orientados a objetos. É pouco adaptável. Atrasa a redução de riscos.



Métodos Formais

Baseado em representações matemáticas. Utilizado em ambientes extremamente complexos e que necessitam de grande robustez e confiabilidade diante da possibilidade de perdas de vidas ou sério prejuízo caso haja falhas. Não são amplamente utilizados no desenvolvimento de software industrial. Métodos formais têm constatado menos erros no software entregue. Exemplo: CleanRoom

Modelo baseado em componentes

Reusabilidade é a palavra que define esse método. Redução de custos de produção e manutenção, entregas mais rápidas e aumento da qualidade. CBSE (Component Based Software Engineering).

O que é um componente? É uma parte do sistema modular, executável, implantável, independente, padronizada, reutilizável que encapsula a implementação e expõe um conjunto de interfaces do sistema.

Os componentes não devem tratar exceções por si mesmos, pois cada aplicação trará seus próprios requisitos para tratamento de exceções.

Modelos iterativos e incrementais

Desenvolver miniprojetos em paralelo e entregar partes diferentes do projeto. Há maior interação com o usuário. Feedback mais intenso. Gerenciamento e mitigação de riscos. Aumenta reúso e qualidade.

Modelos em Prototipagem

Protótipo serve como um mecanismo de identificação de requisitos. É uma forma de entendê-los melhor para posteriormente desenvolver o software. Apropriado para quando o cliente ainda não definiu detalhadamente os requisitos.
Os custos são controlados e as partes interessadas do sistema podem experimentar o protótipo mais cedo no processo de desenvolvimento.
Há dois tipos de desenvolvimento usando o Modelo em Prototipagem:

- **Desenvolvimento Exploratório:** mantém o protótipo
- **Desenvolvimento Throwaway:** joga o protótipo fora

Modelo em Espiral

Foi proposto originalmente, em 1988, por Boehm.
Um processo de software orientado a riscos.

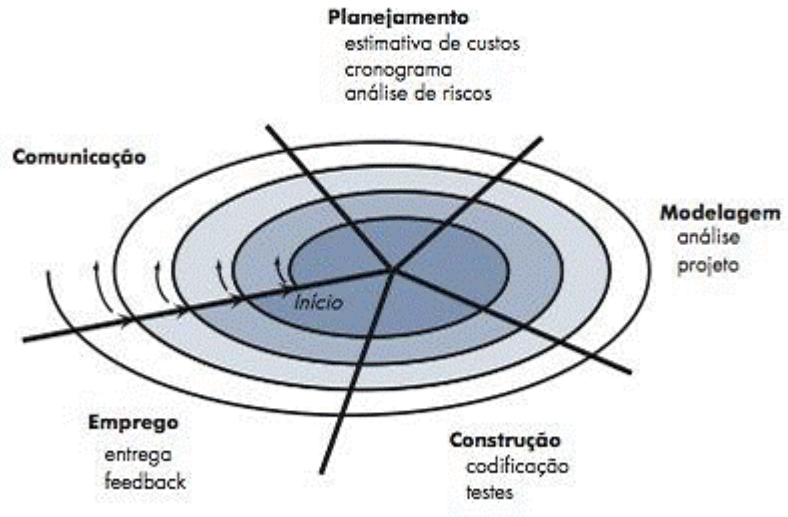
Modelo em cascata + prototipação = Modelo em Espiral

Fases de acordo com Sommerville:

- Determinar objetivos, alternativas e restrições;
- Avaliar as alternativas, identificar e resolver os riscos;
- Desenvolver e testar;
- Planejar as próximas fases;

Fases de acordo com Pressman:

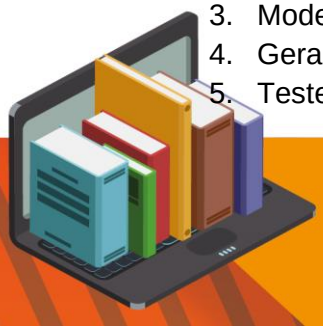
- Comunicação;
- Planejamento (**análise de riscos**, estimativa de custos);
- Modelagem (análise e design);
- Construção (code, test);
- Implantação (entrega, feedback);

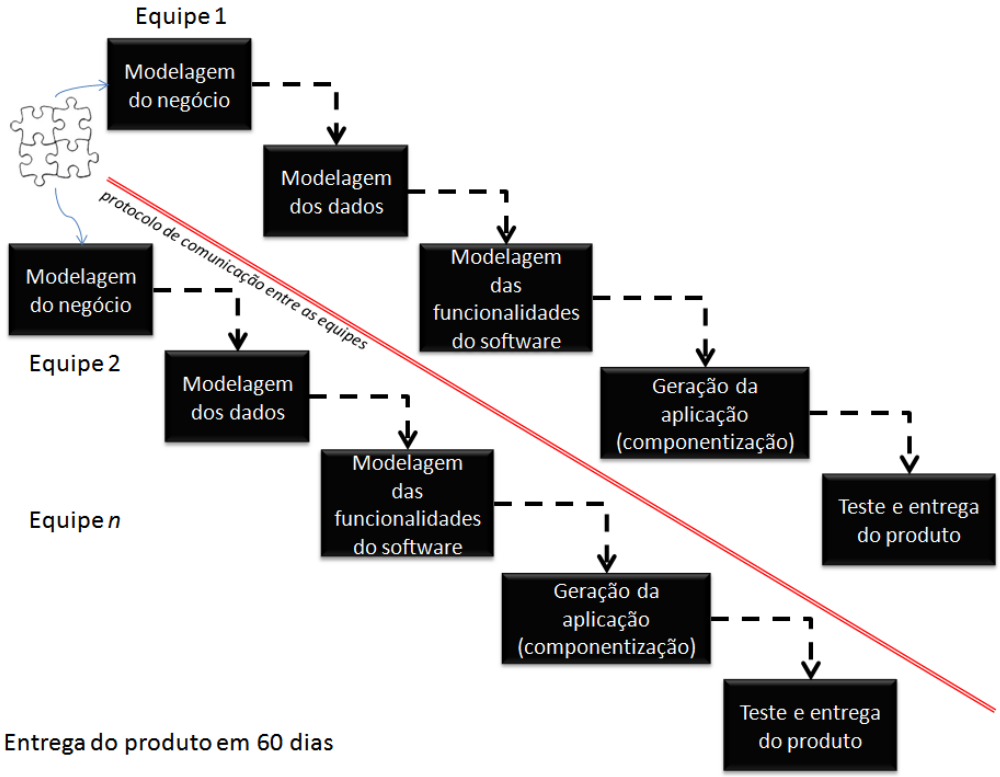


Modelo RAD

O Modelo RAD (Rapid Application Development) é um modelo sequencial linear que enfatiza um ciclo de desenvolvimento extremamente curto. O desenvolvimento rápido é obtido usando uma abordagem de construção baseada em componentes. Os requisitos devem ser bem entendidos. Deve ser possível efetivar a modularização da aplicação. Exige uma equipe maior para conseguir separar as equipes para desenvolver cada módulo independente. É um modelo de desenvolvimento de software iterativo e incremental que enfatiza um ciclo de desenvolvimento extremamente curto (entre 60 e 90 dias).
O processo consiste basicamente em:

1. Modelagem do negócio
2. Modelagem dos dados
3. Modelagem do processo (da funcionalidade)
4. Geração da aplicação
5. Teste e Modificação (Entrega)





Processo Unificado (RUP)

É um framework iterativo e incremental de desenvolvimento de software, centrado na arquitetura, planejado por riscos, guiado por casos de uso e orientado a objetos.
É visto sob 3 perspectivas: Estática, Dinâmica e Prática.
Possui 6 boas práticas, 9 disciplinas e 4 fases.

(mais detalhes no capítulo específico sobre RUP)

Questões
Caso você seja assinante do sistema de questões QConcursos ou do Estratégia Concursos, clique na respectiva imagem abaixo (segure ctrl ao clicar para abrir em outra aba se estiver usando um navegador) e tenha acesso ao caderno de questões referente aos temas acima apresentados para você testar seus conhecimentos.

Qconcursos

Estratégia Concursos