



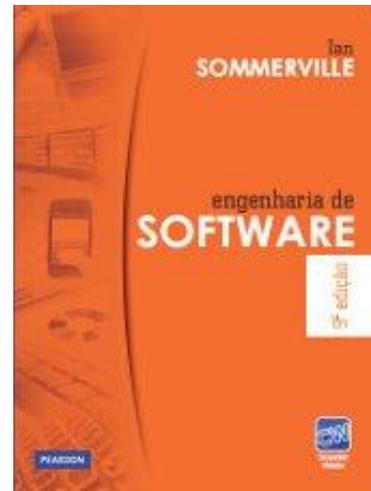
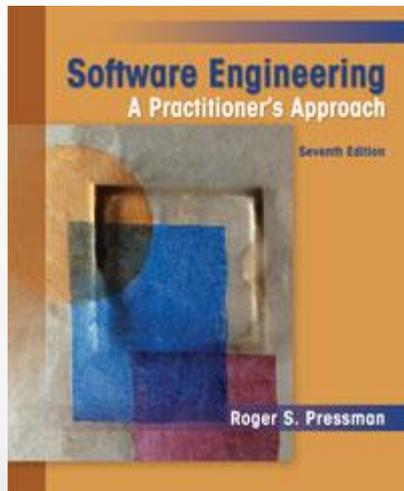
Conceitos Básicos e Ciclo de Vida

Prof. Lúcio Camilo
Email: luciocamilo@gmail.com
<http://www.itnerante.com.br/profile/LucioCamilo>



Bibliografia

- Pressman, Roger S. Software Engineering: A Practitioner's Approach. Editora: McGraw-Hill.
- Sommerville, Ian. Software Engineering. Editora: Addison Wesley.

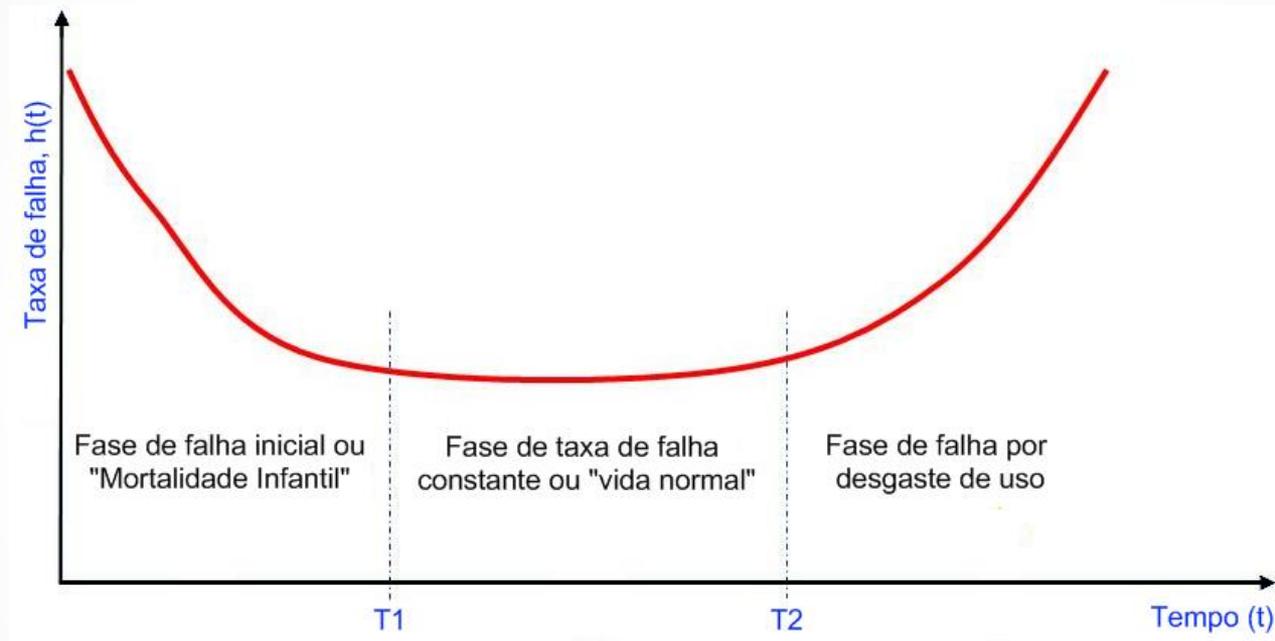


O que é um Software

“Software consiste em instruções que quando executadas fornecem características, funções e desempenho desejados”

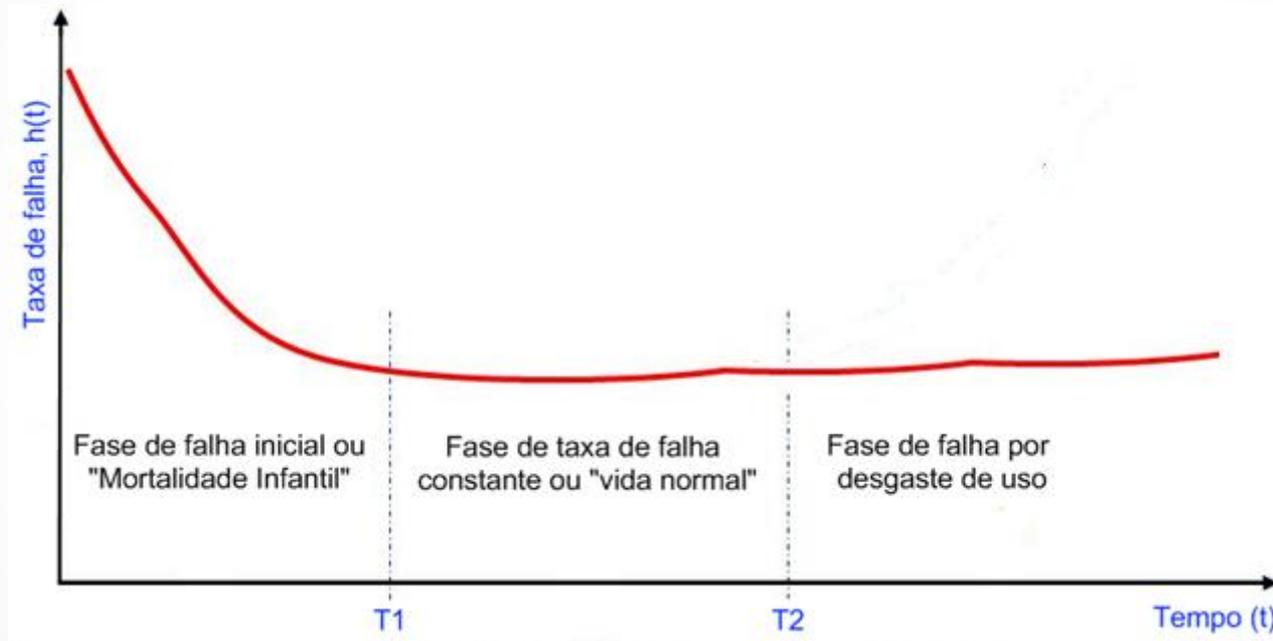


Curva de Defeitos para Hardware

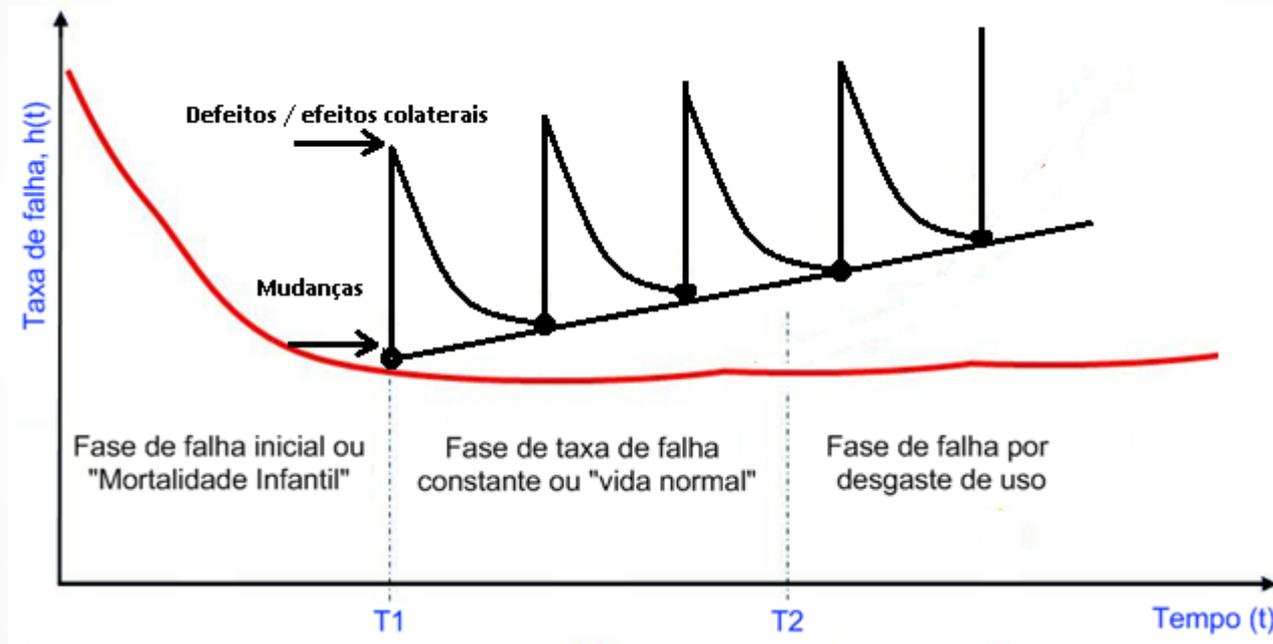


- Hardware se desgasta com o passar do tempo
- Como eu conserto um hardware?

Curva de Defeitos para Software



Curva de Defeitos para Software

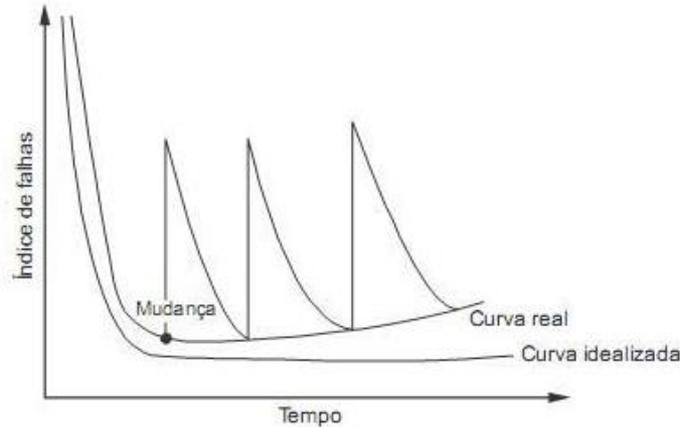


Características do Software

1. Software é desenvolvido, e não fabricado, ou manufaturado;
2. Software não se desgasta, porém se deteriora e sofre constantes modificações;
3. Maior parte dos softwares é desenvolvida de forma personalizada, sob encomenda;

Q01 – FCC - 2012 – TRT6

Considere o gráfico a seguir que mostra as curvas de taxa de falhas, idealizada e real.

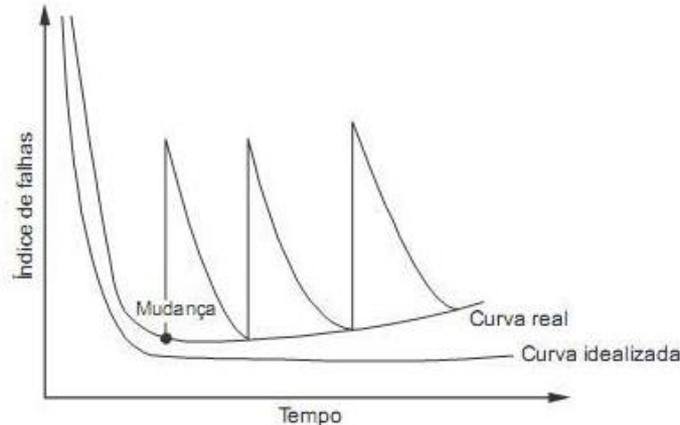


No gráfico, a curva

- idealizada se refere à taxa de falhas do software e considera o processo de manutenção e mudanças do software no seu ciclo de vida.
- idealizada e a curva real se referem à taxa de falhas do software. A curva real mostra que as sucessivas alterações do software tendem a introduzir mais erros antes da estabilização dos erros de alterações anteriores, ocasionando a tendência crescente do índice de falhas.
- idealizada se refere ao hardware, que tem um alto índice de falhas no início do seu ciclo de vida ocasionadas por defeitos de fabricação e projeto, mas depois se estabiliza até o final da vida do produto.
- real se refere ao hardware que, desde o início do ciclo de vida do produto, pode apresentar problemas relacionados ao envelhecimento, acúmulo de poeira, vibração, abuso, temperaturas extremas, entre outros.
- idealizada e a curva real se referem à taxa de falhas do hardware, produto lógico, que não se desgasta, mas se deteriora rapidamente em função da introdução de erros oriundos de atividades de manutenção.

Q01 – FCC - 2012 – TRT6

Considere o gráfico a seguir que mostra as curvas de taxa de falhas, idealizada e real.



No gráfico, a curva

- a) idealizada se refere à taxa de falhas do software e considera o processo de manutenção e mudanças do software no seu ciclo de vida.
- ➔ b) idealizada e a curva real se referem à taxa de falhas do software. A curva real mostra que as sucessivas alterações do software tendem a introduzir mais erros antes da estabilização dos erros de alterações anteriores, ocasionando a tendência crescente do índice de falhas.
- c) idealizada se refere ao hardware, que tem um alto índice de falhas no início do seu ciclo de vida ocasionadas por defeitos de fabricação e projeto, mas depois se estabiliza até o final da vida do produto.
- d) real se refere ao hardware que, desde o início do ciclo de vida do produto, pode apresentar problemas relacionados ao envelhecimento, acúmulo de poeira, vibração, abuso, temperaturas extremas, entre outros.
- e) idealizada e a curva real se referem à taxa de falhas do hardware, produto lógico, que não se desgasta, mas se deteriora rapidamente em função da introdução de erros oriundos de atividades de manutenção.

Mitos do Software (Gerenciamento)

Mito: Já temos um manual repleto de padrões e procedimentos para a construção de software. Isso não oferecerá tudo o que eles precisam saber?

Realidade: O manual é usado? Todos sabem que ele existe? Ele é completo? É adaptável? Está alinhado para melhorar o tempo de entrega, mantendo o foco na qualidade?

Mito: Se o cronograma atrasar, poderemos acrescentar mais programadores e ficarmos em dia.

Realidade: Desenvolvimento de software não é um processo mecânico como o de fábrica. “Acrescentar pessoas num projeto de software atrasado só o tornará mais atrasado ainda”. Pode-se adicionar pessoas, porém somente de forma planejada e bem coordenada.

Mito: Se eu decidir terceirizar o projeto de software, posso relaxar e deixar essa empresa realizá-lo.

Realidade: Se uma organização não souber gerenciar projetos, irá enfrentar dificuldades ao terceirizá-los.

Mitos do Software (Clientes)

Mito: Uma definição geral dos objetivos é suficiente para começar a escrever os programas. Podemos preencher detalhes posteriormente.

Realidade: Nem sempre é possível ter uma definição ampla e estável dos requisitos, porém requisitos ambíguos são a fórmula para o fracasso. Uma definição não ambígua ocorre pela comunicação contínua e eficaz.

Mito: Os requisitos de software mudam continuamente, mas as mudanças podem ser facilmente assimiladas, pois o software é flexível.

Realidade: Os requisitos de software mudam, mas o impacto da mudança varia dependendo do momento em que ela foi introduzida. Quanto mais tarde uma mudança for necessária, maior o custo.

Custo das Mudanças

Fases	Custo de Manutenção
Definição	1x
Desenvolvimento	1.5 – 6x
Manutenção	60 – 100x

Mitos do Software (Profissionais)

Mito: Uma vez feito o programa e colocado em uso, nosso trabalho está terminado.

Realidade: “Quanto antes começar a codificar, mais tempo levará para terminá-lo”. Levantamentos indicam que entre 60 e 80% de todo o esforço será após a entrega do software pela primeira vez.

Mito: Até que o programa entre em funcionamento, não há maneira de avaliar sua qualidade.

Realidade: Garantia de qualidade pode ser aplicado desde a concepção de um projeto, através da revisão técnica.

Mito: O único produto passível de entrega é o programa em funcionamento.

Realidade: Um programa funcional é apenas parte de uma configuração que inclui muitos elementos.

Mitos do Software (Profissionais)

Mito: A engenharia de software nos fará criar documentação volumosa e desnecessária, e irá nos atrasar.

Realidade: A engenharia de software não trata de criação de documentos, trata da criação de um produto de qualidade.

Fatos Reais

1. Deve-se fazer um esforço concentrado para compreender o problema antes de desenvolver uma solução de software;
2. Projetar é uma atividade fundamental. Requisitos são cada vez mais complexos devido a sofisticação e variação de ambiente;
3. Software deve apresentar qualidade elevada. Algumas vezes uma falha de software pode causar uma catástrofe;
4. Software deve ser passível de manutenção.

O que é Engenharia de Software?

“Engenharia de software é o estabelecimento e o emprego de sólidos princípios de engenharia de modo a obter software de maneira econômica, que seja confiável e funcione de forma eficiente em máquinas reais”

Fritz Bauer

“Aplicação de uma abordagem sistemática, disciplinada e quantificável no desenvolvimento, na operação e na manutenção de software.”

IEEE Standard Glossary of Software Engineering

“Engenharia de Software envolve a aplicação prática de conhecimento científico para o projeto e construção de programas de computador e a documentação associada necessária para desenvolvê-los, operá-los e mantê-los.”

Barry Boehm

Objetivos

- Obter software de qualidade;
- Produtividade no desenvolvimento, operação e manutenção do software;
- Controlar os custos, prazos e qualidades;
- Proporcionar o melhor custo benefício entre qualidade e produtividade
- Resultados de qualidade dentro do cronograma
- Vantagens:
 - Capacidade de produzir sistemas confiáveis economicamente e de forma rápida;
 - Geralmente é mais barato a longo prazo usar métodos e técnicas do que simplesmente escrever código.

Princípios

1. Razão de Existir
 - “Gerar valor aos seus usuários”
2. KISS (Keep It Simple, Stupid!)
 - Simplicidade
3. Mantenha a Visão
 - Direcionamento baseado na arquitetura
4. O que um Produz, Outros Consomem
 - Pense sempre que alguém mais irá precisar
5. Aberto para o Futuro
 - Adaptável a mudanças
6. Planeje com Antecedência, Visando a Reutilização
 - Reuso é fundamental
7. Pense!

Engenharia de Software = Engenharia de Sistemas ?

- Engenharia de Sistemas se preocupa com todos os aspectos de sistemas baseados em computador, e não apenas Software.
- Engloba:
 - Software,
 - Hardware,
 - Processos,
 - Pessoas e outros sistemas.
- Engenharia de sistemas é algo maior, que contempla a engenharia de software

Abordagem em Camadas



Abordagem em Camadas

- Foco na Qualidade:
 - Qualquer abordagem deve estar fundamentada em um comprometimento com a qualidade.
- Processos:
 - O processo na engenharia de software é o que permite o desenvolvimento de software de forma racional e dentro do prazo.
 - Define uma metodologia que deve ser estabelecida.
 - Estabelece o contexto em que são aplicados os métodos técnicos e os marcos.
- Métodos:
 - Fornecem as informações técnicas para desenvolver software.
 - Inclui comunicação, análise de requisitos, , modelagem, construção, testes e suporte.
- Ferramentas:
 - Suporte automatizado ou semiautomatizado para o processo e para os métodos.

A engenharia de software define quatro fases para o ciclo de vida de um sistema: iniciação, elaboração, construção e transição, sendo essa última responsável pela homologação dos artefatos junto ao cliente.

Certo

Errado

A engenharia de software define quatro fases para o ciclo de vida de um sistema: iniciação, elaboração, construção e transição, sendo essa última responsável pela homologação dos artefatos junto ao cliente.

Certo

➡ Errado

Engenharia de software não está relacionada somente aos processos técnicos de desenvolvimento de softwares, mas também a atividades como gerenciamento de projeto e desenvolvimento de ferramentas, métodos e teorias que apoiem a produção de softwares.

Certo

Errado

Engenharia de software não está relacionada somente aos processos técnicos de desenvolvimento de softwares, mas também a atividades como gerenciamento de projeto e desenvolvimento de ferramentas, métodos e teorias que apoiem a produção de softwares.

→ Certo

Errado

A engenharia de *software* engloba processos, métodos e ferramentas. Um de seus focos é a produção de *software* de alta qualidade a custos adequados.

Certo

Errado

A engenharia de *software* engloba processos, métodos e ferramentas. Um de seus focos é a produção de *software* de alta qualidade a custos adequados.

➔ Certo

Errado

A Engenharia de Software

- a) é uma área da computação que visa abordar de modo sistemático as questões técnicas e não técnicas no projeto, implantação, operação e manutenção no desenvolvimento de um software.
- b) consiste em uma disciplina da computação que aborda assuntos relacionados a técnicas para a otimização de algoritmos e elaboração de ambientes de desenvolvimento.
- c) trata-se de um ramo da TI que discute os aspectos técnicos e empíricos nos processos de desenvolvimento de sistemas, tal como a definição de artefatos para a modelagem ágil.
- d) envolve um conjunto de itens que abordam os aspectos de análise de mercado, concepção e projeto de software, sendo independente da engenharia de um sistema.
- e) agrupa as melhores práticas para o concepção, projeto, operação e manutenção de artefatos que suportam a execução de programas de computador, tais como as técnicas de armazenamento e as estruturas em memória principal.

A Engenharia de Software

- ➔ a) é uma área da computação que visa abordar de modo sistemático as questões técnicas e não técnicas no projeto, implantação, operação e manutenção no desenvolvimento de um software.
- b) consiste em uma disciplina da computação que aborda assuntos relacionados a técnicas para a otimização de algoritmos e elaboração de ambientes de desenvolvimento.
- c) trata-se de um ramo da TI que discute os aspectos técnicos e empíricos nos processos de desenvolvimento de sistemas, tal como a definição de artefatos para a modelagem ágil.
- d) envolve um conjunto de itens que abordam os aspectos de análise de mercado, concepção e projeto de software, sendo independente da engenharia de um sistema.
- e) agrupa as melhores práticas para o concepção, projeto, operação e manutenção de artefatos que suportam a execução de programas de computador, tais como as técnicas de armazenamento e as estruturas em memória principal.

Modelos de Ciclo de Vida

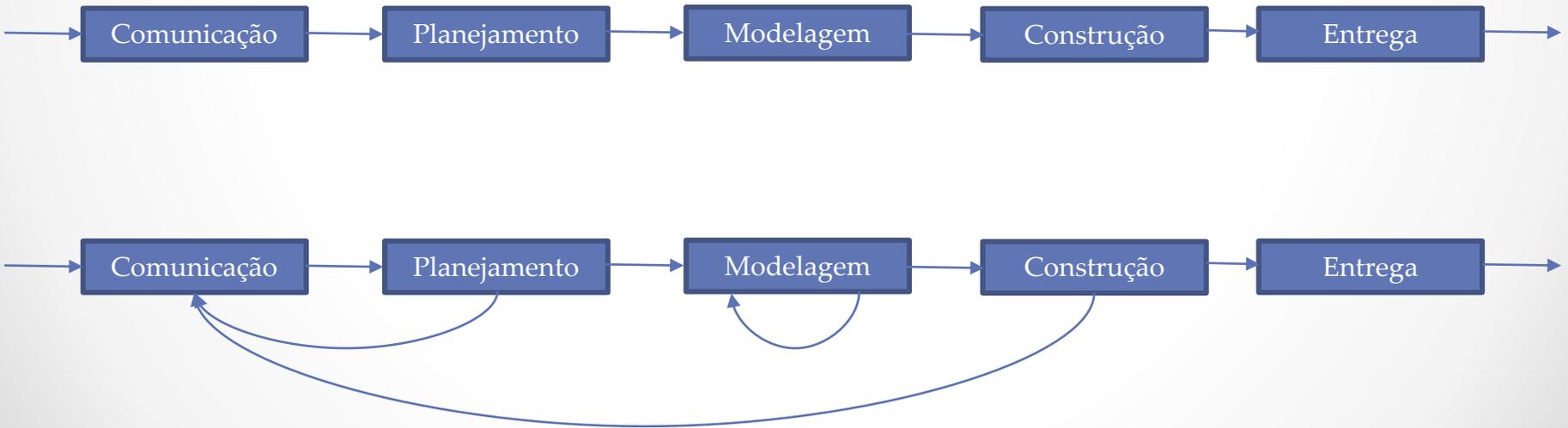
- Conjunto de atividades para construção do Software
- Escolhidos de acordo com as características do projeto
- “Representação abstrata e simplificada do processo de desenvolvimento de software a partir de uma perspectiva específica.”

Tipos de Modelos

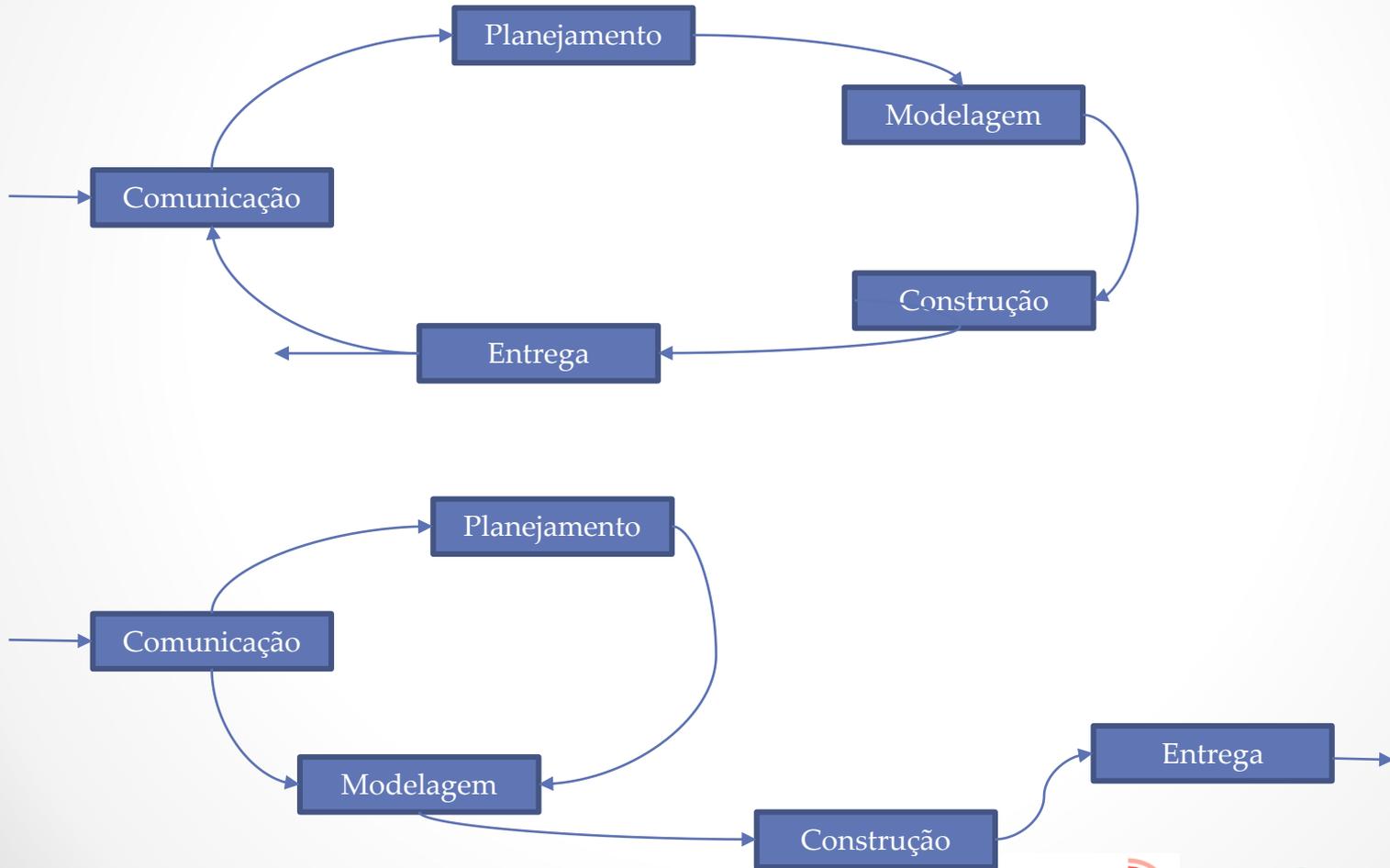
- Modelos Tradicionais (Prescritivos)
 - Modelo Cascata
 - Incremental
 - Incremental
 - RAD
 - Evolucionário (Iterativos)
 - Prototipação
 - Espiral
 - Concorrentes
- Modelos Especializados
 - Baseado em Componentes
 - Métodos Formais
 - Orientado a Aspectos

Metodologia Genérica

- Comunicação
- Planejamento
- Modelagem
- Construção
- Entrega



Metodologia Genérica

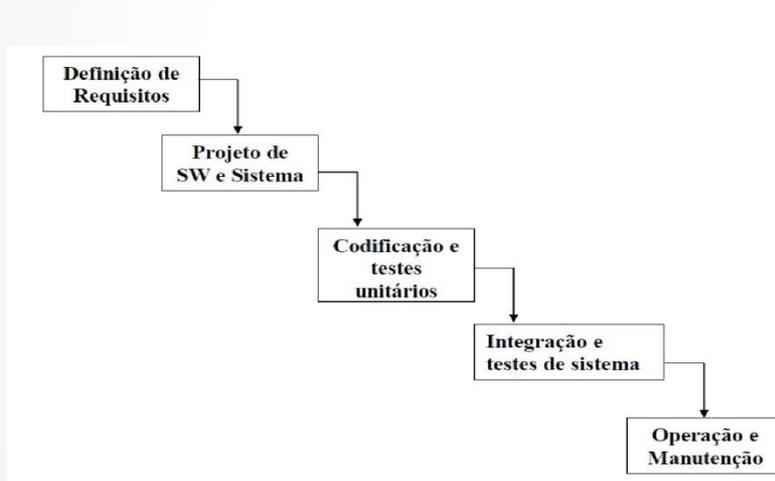


Modelo Cascata

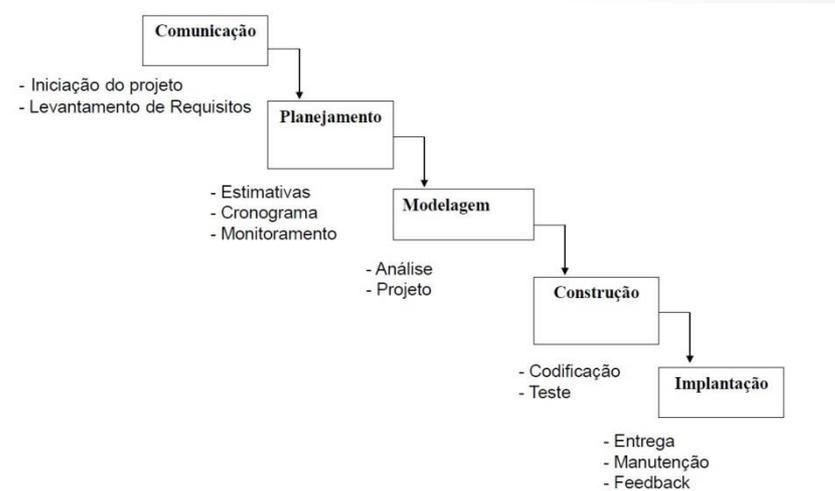
- Clássico, Linear, Sequencial;
- Requisitos de um problema devem ser estáveis e bem compreendidos;
- Trabalho flui da Comunicação ao Emprego de forma linear;
- Abordagem sequencial e sistemática

Modelo Cascata

Sommerville



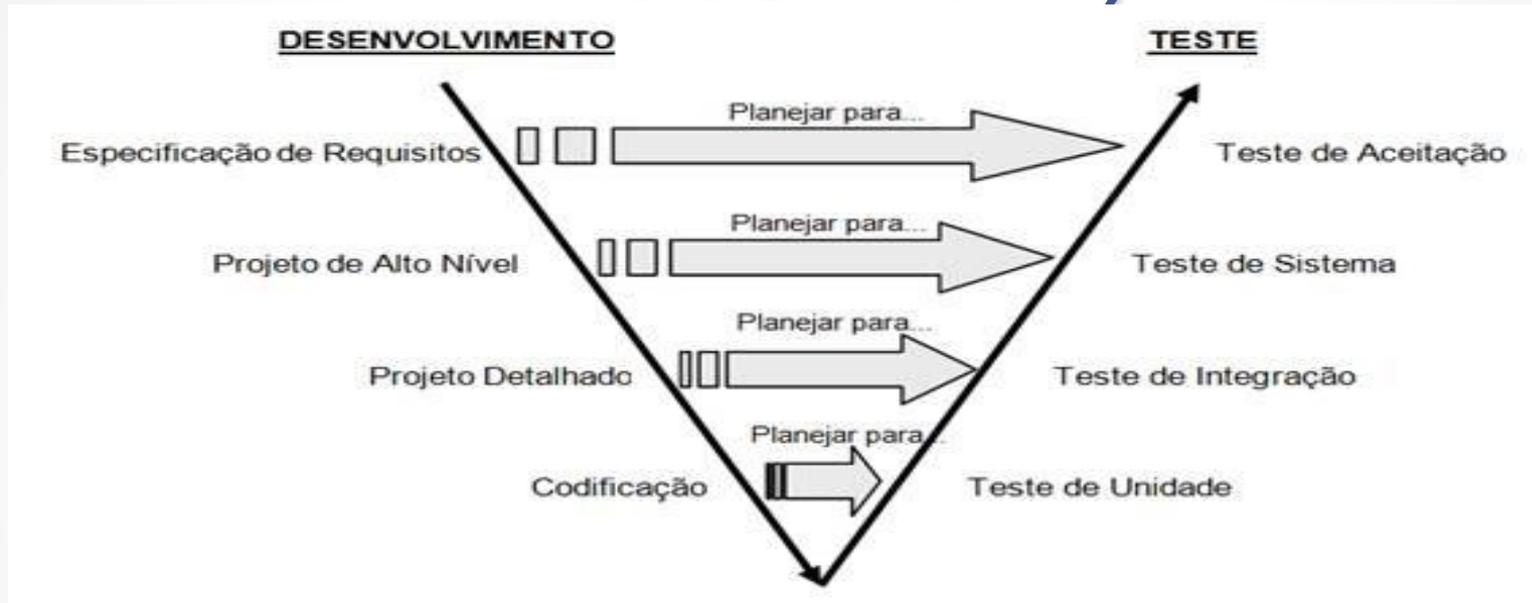
Pressman



Modelo Cascata

- Vantagens:
 - Simples e de fácil gerência
 - Adequado para sistemas de baixa complexidade
- Problemas:
 - Projetos reais, geralmente não seguem um fluxo sequencial
 - Geralmente o cliente não sabe definir todos os requisitos no início
 - Uma versão funcional somente estará disponível ao final do projeto
 - Erros são detectados no final
 - Sistema é entregue todo de uma vez
 - Uma etapa depende da outra (Equipes ociosas)
 - Atrasa a redução de riscos

Modelo V (Sétima Edição Preesman)



- Variação do Modelo Cacata
- Descreve a relação entre ações de garantia da qualidade e as ações do ciclo de vida.
- Oferece uma forma para visualizar como as ações de qualidade podem ser aplicadas ao processo de trabalho.

O modelo mais tradicional de desenvolvimento de software é o modelo em cascata. Considerando a utilização desse modelo e suas fases, assinale a alternativa que apresenta uma afirmação verdadeira.

- (A) O primeiro estágio de desenvolvimento de um novo sistema consiste na definição de requisitos.
- (B) A divisão dos requisitos para implementação do sistema em hardware ou software é feita na fase de operação e manutenção.
- (C) A especificação do sistema é produzida após o estágio de implementação e teste de unidade.
- (D) A integração e o teste dos programas individuais são feitos no estágio de implementação e teste de unidade.
- (E) Não há necessidade de se produzir qualquer tipo de documentação em suas fases.

O modelo mais tradicional de desenvolvimento de software é o modelo em cascata. Considerando a utilização desse modelo e suas fases, assinale a alternativa que apresenta uma afirmação verdadeira.

- ➔ (A) O primeiro estágio de desenvolvimento de um novo sistema consiste na definição de requisitos.
- (B) A divisão dos requisitos para implementação do sistema em hardware ou software é feita na fase de operação e manutenção.
- (C) A especificação do sistema é produzida após o estágio de implementação e teste de unidade.
- (D) A integração e o teste dos programas individuais são feitos no estágio de implementação e teste de unidade.
- (E) Não há necessidade de se produzir qualquer tipo de documentação em suas fases.

As fases do modelo de ciclo de vida em cascata são demonstração, análise de requisitos, pré-projeto, desenvolvimento, projeto básico, homologação, implantação e manutenção, respectivamente.

Certo

Errado

As fases do modelo de ciclo de vida em cascata são demonstração, análise de requisitos, pré-projeto, desenvolvimento, projeto básico, homologação, implantação e manutenção, respectivamente.

Certo

➡ Errado

O modelo de ciclo de vida em cascata é considerado o paradigma mais antigo da engenharia de software. Apesar de apresentar diversas desvantagens em relação ao modelo incremental, pode ser útil principalmente em situações que :

- a) a equipe de desenvolvimento é grande.
- b) os requisitos são fixos.
- c) o prazo é curto.
- d) o cliente é indeciso.
- e) os requisitos são instáveis.

O modelo de ciclo de vida em cascata é considerado o paradigma mais antigo da engenharia de software. Apesar de apresentar diversas desvantagens em relação ao modelo incremental, pode ser útil principalmente em situações que :

- a) a equipe de desenvolvimento é grande.
- ➔ b) os requisitos são fixos.
- c) o prazo é curto.
- d) o cliente é indeciso.
- e) os requisitos são instáveis.

Flávio pretende desenvolver um software seguindo os estágios do modelo em cascata proposto por Sommerville, em razão de ponderações que faz em relação a outros modelos quanto à solução de um problema que se apresenta. Desta forma ele definiu em seu cronograma, na ordem apresentada pelo autor, as seguintes etapas do ciclo de vida de software:

- a) Projeto de sistema e software; Definição de requisitos; Implementação e teste de unidade; Integração e teste de sistema; Operação e manutenção.
- b) Projeto de sistema e software; Engenharia de software; Integração e teste de sistema; Análise de requisitos funcionais e técnicos; Operação e manutenção; Implementação e teste de unidade.
- c) Projeto de sistema e software; Análise de requisitos; Engenharia de requisitos; Implantação; Testes de sistemas; Operação e manutenção.
- d) Definição de requisitos; Engenharia de requisitos; Integração e teste de sistema; Projeto de sistema e software; Implementação e teste de unidade; Operação e manutenção; Integração e teste de sistema.
- e) Definição de requisitos; Projeto de sistema e software; Implementação e teste de unidade; Integração e teste de sistema; Operação e manutenção.

Flávio pretende desenvolver um software seguindo os estágios do modelo em cascata proposto por Sommerville, em razão de ponderações que faz em relação a outros modelos quanto à solução de um problema que se apresenta. Desta forma ele definiu em seu cronograma, na ordem apresentada pelo autor, as seguintes etapas do ciclo de vida de software:

- a) Projeto de sistema e software; Definição de requisitos; Implementação e teste de unidade; Integração e teste de sistema; Operação e manutenção.
- b) Projeto de sistema e software; Engenharia de software; Integração e teste de sistema; Análise de requisitos funcionais e técnicos; Operação e manutenção; Implementação e teste de unidade.
- c) Projeto de sistema e software; Análise de requisitos; Engenharia de requisitos; Implantação; Testes de sistemas; Operação e manutenção.
- d) Definição de requisitos; Engenharia de requisitos; Integração e teste de sistema; Projeto de sistema e software; Implementação e teste de unidade; Operação e manutenção; Integração e teste de sistema.
- ➔ e) Definição de requisitos; Projeto de sistema e software; Implementação e teste de unidade; Integração e teste de sistema; Operação e manutenção.

Q10 – QUADRIX - 2014 – SERPRO

O Modelo Cascata é um modelo de processo clássico utilizado na indústria de software. Esse modelo:

- I. prevê uma atividade de revisão ao final de cada fase, para que se avalie se o projeto pode passar à fase seguinte.
- II. funciona bem com projetos nos quais os requisitos não são bem conhecidos e estáveis.
- III. funciona bem quando a preocupação com custo ou tempo de desenvolvimento está acima da preocupação com a qualidade.
- IV. é iterativo e incremental, dividindo o desenvolvimento de software nas fases de levantamento de requisitos, análise, projeto, codificação, teste e operação.

Está correto o que se afirma em:

- a) I, somente.
- b) II e IV, somente
- c) II e III, somente.
- d) III e IV, somente
- e) I e II, somente

Q10 – QUADRIX - 2014 – SERPRO

O Modelo Cascata é um modelo de processo clássico utilizado na indústria de software. Esse modelo:

- I. prevê uma atividade de revisão ao final de cada fase, para que se avalie se o projeto pode passar à fase seguinte.
- II. funciona bem com projetos nos quais os requisitos não são bem conhecidos e estáveis.
- III. funciona bem quando a preocupação com custo ou tempo de desenvolvimento está acima da preocupação com a qualidade.
- IV. é iterativo e incremental, dividindo o desenvolvimento de software nas fases de levantamento de requisitos, análise, projeto, codificação, teste e operação.

Está correto o que se afirma em:

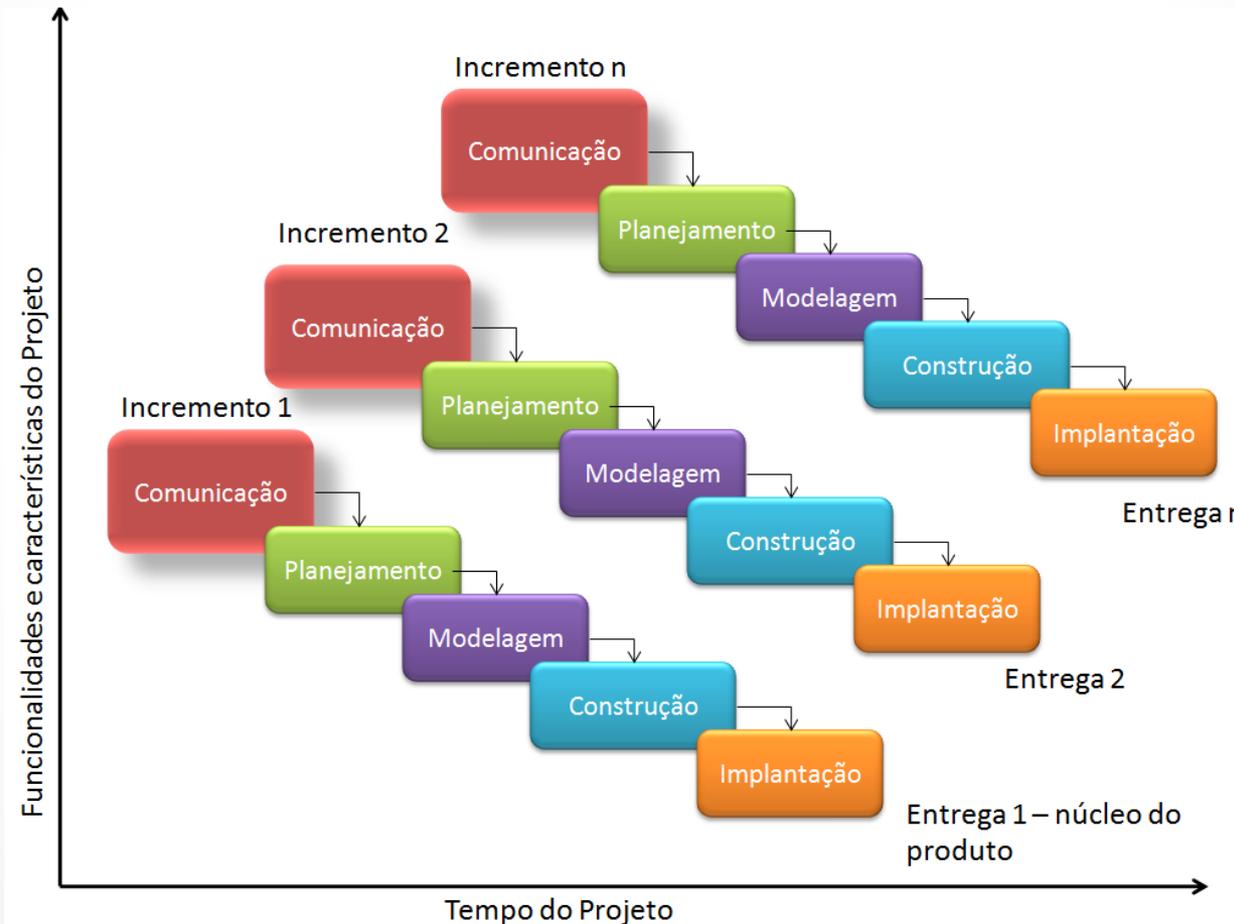
-  a) I, somente.
b) II e IV, somente
c) II e III, somente.
d) III e IV, somente
e) I e II, somente

Modelo Incremental

- Requisitos bem definidos, porém é necessário fornecer uma versão funcional em pouco tempo
- Vários incrementos até ficar pronto
- Foco na entrega de um produto operacional em cada incremento
- Permite administrar riscos técnicos

- Vantagens:
 - Datas inflexíveis
 - Problemas Complexos
- Desvantagens:
 - Projetos grandes precisam de muitas equipes
 - Sistemas não modularizados são problemáticos

Modelo Incremental

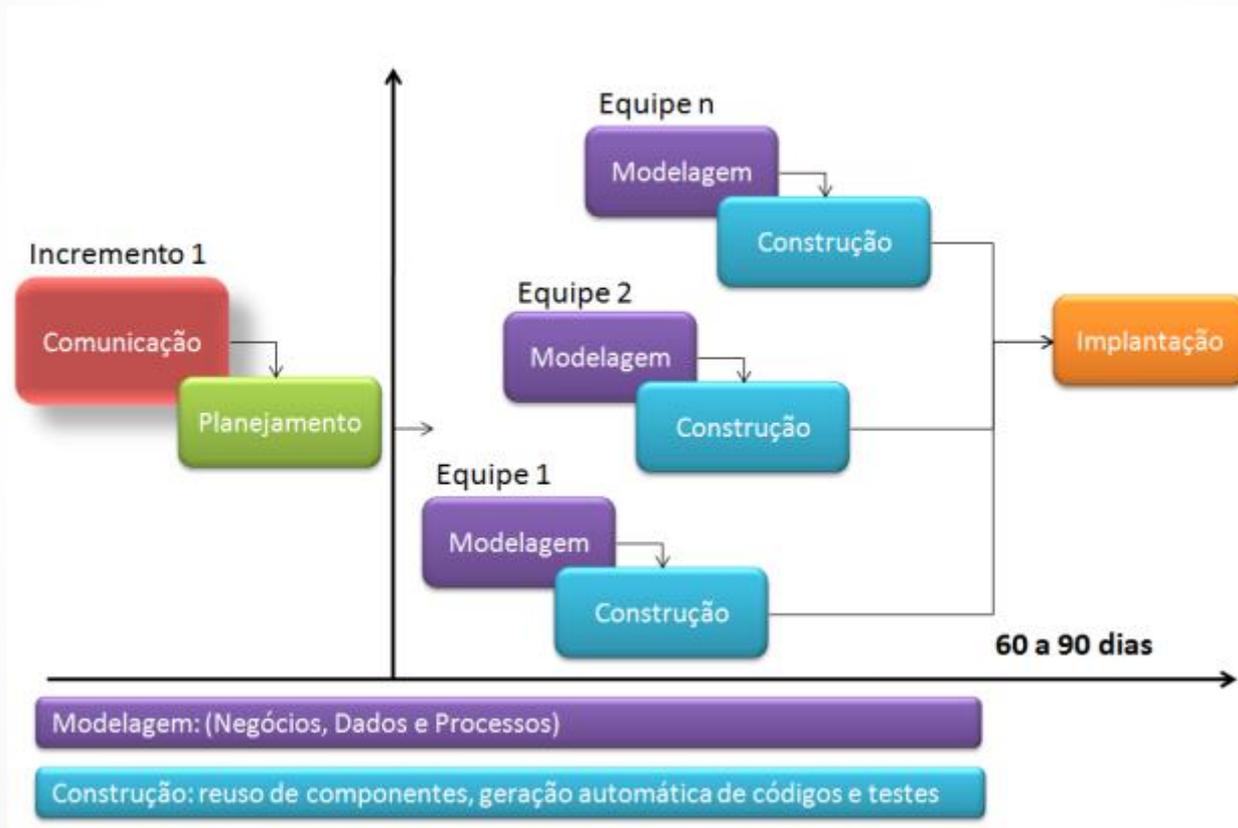


Modelo RAD (Sexta Edição – Pressman)

- *Rapid Application Development*
- Enfatiza o uso de componentes
- Equipes trabalham em paralelo
- Sistema funcional em 60–90 dias
- Exige modularidade

- Vantagens:
 - Uso de Componentes
 - Redução do Tempo
- Desvantagens:
 - Necessidade de modularização

Modelo RAD



Modelo Evolucionários

- Software evolui com o tempo
- Requisitos são definidos e depois refinados
- Evolui-se a cada iteração
- Uma iteração é uma passada por todas as fases

- Tipos:
 - Prototipação
 - Espiral

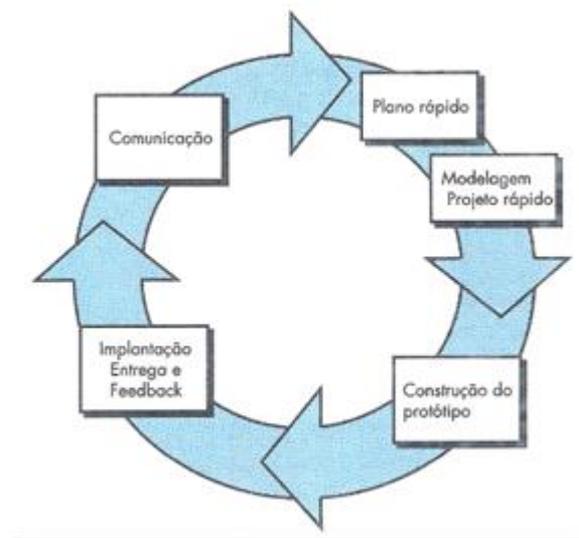
Prototipação

- Mecanismo para identificar requisitos de software
- Utilizado para entender requisitos
- Adequado para quando o cliente não tem detalhes dos requisitos

- Prototipagem Evolucionária:
 - Protótipo segue em evolução até tornar-se o sistema real
- Prototipagem Descartável:
 - Começa com os requisitos mais difíceis e menos compreendidos
 - Descarta-se o protótipo ao final e a implementação continua
 - Podem ser aplicados no contexto de qualquer modelo

Prototipação

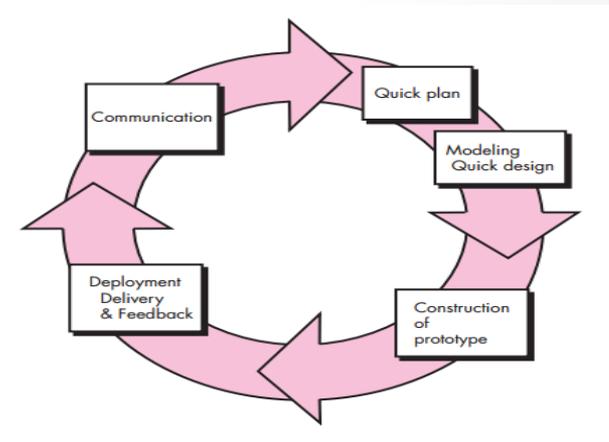
- Problemas:
 - Clientes querem o protótipo como sistema
 - Escolhas de desenvolvimento precipitadas



- Foco na representação dos aspectos de software que serão visíveis aos usuários finais.

Prototipação

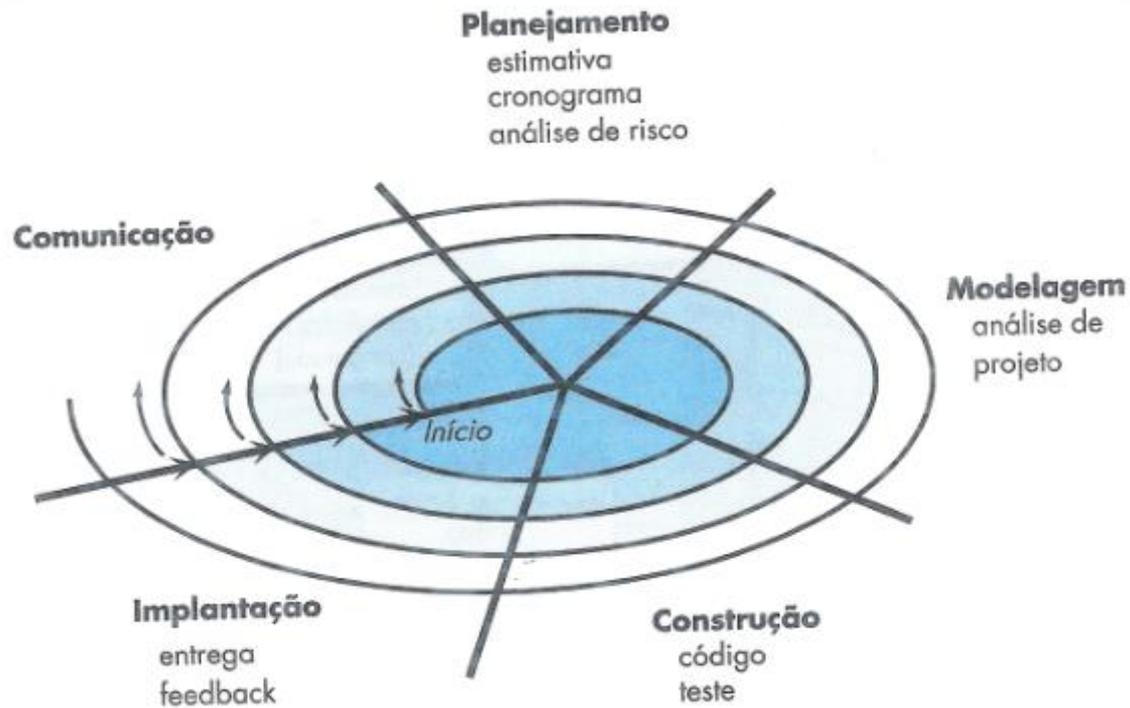
- Comunicação
 - Definir objetivos gerais
- Projeto Rápido
 - Planejamento,
 - Modelagem e
 - Construção de um protótipo
- Entrega e Feedback
 - Ajustes eventuais



Espiral

- Abordagem cíclica guiada por riscos
- Aumenta incrementalmente o grau de definição e implementação de um sistema
- Pontos de Controle
- Combina a natureza iterativa com aspectos sistemáticos do cascata
- Vantagens:
 - Adaptabilidade
 - Redução do Risco (através da prototipação)
- Desvantagens:
 - A cada ciclo o custo será revisado
 - Especialização na avaliação de riscos

Espiral



Modelo Concorrente

- Enfatiza os estados associados a cada tarefa
- Equipe envolvida em várias atividades simultâneas
- Permite representar elementos concorrentes de qualquer modelo
- Eventos sinalizam alterações de estados



Q11 - FCC - 2014 – TRT13 – Análise de Sistemas

O desenvolvimento evolucionário baseia-se na ideia de desenvolvimento de uma implementação inicial, expondo o resultado aos comentários do usuário e refinando-o em novas versões até que seja desenvolvido um sistema adequado. As atividades de especificação, desenvolvimento e validação são intercaladas ao invés de separadas, com rápido feedback entre elas.

Sommerville define dois tipos fundamentais de desenvolvimento evolucionário. Considere:

- I. Descrever todos os requisitos não funcionais antes de fazer o protótipo. Descrever os requisitos funcionais e técnicos. Implementar todos requisitos e desenvolver novo protótipo.
- II. Trabalhar com o cliente para explorar os requisitos e entregar um sistema final. O desenvolvimento começa com as partes do sistema compreendidas. O sistema evolui por meio da adição de novas características propostas pelo cliente.
- III. Incorporar e implementar todas as mudanças do software no primeiro estágio do desenvolvimento, definindo todos os requisitos técnicos. Formar um protótipo a partir daí. O sistema evolui por meio da adição de novas características propostas pelo cliente.
- IV. Compreender os requisitos do cliente e, a partir disso, desenvolver melhor definição de requisitos para o sistema. O protótipo se concentra na experimentação dos requisitos mal compreendidos do cliente.

De acordo com Sommerville

- a) I é Prototipação throwaway; III é Prototipação incremental.
- b) I é Prototipação incremental; IV é Desenvolvimento exploratório.
- c) II é Desenvolvimento exploratório; IV é Prototipação throwaway.
- d) III é Desenvolvimento exploratório; IV é Prototipação revisional.
- e) II é Prototipação exploratória; III é Prototipação throwaway.

Q11 - FCC - 2014 – TRT13 – Análise de Sistemas

O desenvolvimento evolucionário baseia-se na ideia de desenvolvimento de uma implementação inicial, expondo o resultado aos comentários do usuário e refinando-o em novas versões até que seja desenvolvido um sistema adequado. As atividades de especificação, desenvolvimento e validação são intercaladas ao invés de separadas, com rápido feedback entre elas.

Sommerville define dois tipos fundamentais de desenvolvimento evolucionário. Considere:

- I. Descrever todos os requisitos não funcionais antes de fazer o protótipo. Descrever os requisitos funcionais e técnicos. Implementar todos requisitos e desenvolver novo protótipo.
- II. Trabalhar com o cliente para explorar os requisitos e entregar um sistema finalO desenvolvimento começa com as partes do sistema compreendidas. O sistema evolui por meio da adição de novas características propostas pelo cliente.
- III. Incorporar e implementar todas as mudanças do software no primeiro estágio do desenvolvimento, definindo todos os requisitos técnicos. Formar um protótipo a partir daí. O sistema evolui por meio da adição de novas características propostas pelo cliente.
- IV. Compreender os requisitos do cliente e, a partir disso, desenvolver melhor definição de requisitos para o sistema. O protótipo se concentra na experimentação dos requisitos mal compreendidos do cliente.

De acordo com Sommerville

- a) I é Prototipação throwaway; III é Prototipação incremental.
- b) I é Prototipação incremental; IV é Desenvolvimento exploratório.
- c) II é Desenvolvimento exploratório; IV é Prototipação throwaway.
- d) III é Desenvolvimento exploratório; IV é Prototipação revisional.
- e) II é Prototipação exploratória; III é Prototipação throwaway.

Q12 - FCC - 2014 – TRT16 – Análise de Sistemas

Os modelos de processo são uma representação abstrata de um processo de software, que podem ser usados para explicar diferentes abordagens para o desenvolvimento de sistemas. Analise as seguintes abordagens:

Desenvolvimento I intercala as atividades de especificação, desenvolvimento e validação. Um sistema inicial é desenvolvido rapidamente baseado em especificações abstratas e depois é refinado com as entradas do cliente para produzir um produto que o satisfaça.

Modelo II considera as atividades fundamentais do processo, compreendendo especificação, desenvolvimento, validação e evolução e as representa como fases de processo separadas, tais como especificação de requisitos, projeto de software, implementação, teste etc.

III baseia-se na existência de um número significativo de partes reusáveis. O processo de desenvolvimento do sistema enfoca a integração destas partes, ao invés de desenvolvê-las a partir do zero.

Os modelos de processo genéricos descritos em I, II e III são, correta e respectivamente, associados a:

- a) em Espiral – Baseado em Componentes – RAD
- b) Evolucionário – em Cascata – Baseado em Componentes
- c) Baseado em Componentes – Sequencial – Refactoring
- d) Ágil – Sequencial – Unified Process
- e) em Cascata – Ágil – Refactoring

Q12 - FCC - 2014 – TRT16 – Análise de Sistemas

Os modelos de processo são uma representação abstrata de um processo de software, que podem ser usados para explicar diferentes abordagens para o desenvolvimento de sistemas. Analise as seguintes abordagens:

Desenvolvimento I intercala as atividades de especificação, desenvolvimento e validação. Um sistema inicial é desenvolvido rapidamente baseado em especificações abstratas e depois é refinado com as entradas do cliente para produzir um produto que o satisfaça.

Modelo II considera as atividades fundamentais do processo, compreendendo especificação, desenvolvimento, validação e evolução e as representa como fases de processo separadas, tais como especificação de requisitos, projeto de software, implementação, teste etc.

III baseia-se na existência de um número significativo de partes reusáveis. O processo de desenvolvimento do sistema enfoca a integração destas partes, ao invés de desenvolvê-las a partir do zero.

Os modelos de processo genéricos descritos em I, II e III são, correta e respectivamente, associados a:

- a) em Espiral – Baseado em Componentes – RAD
- ➔ b) Evolucionário – em Cascata – Baseado em Componentes
- c) Baseado em Componentes – Sequencial – Refactoring
- d) Ágil – Sequencial – Unified Process
- e) em Cascata – Ágil – Refactoring

Q13 – FCC - 2015 – TRE/RR

Deve-se propor um modelo e processo(s) de software adequado(s) para o desenvolvimento de sistemas com as seguintes características:

- I. Os requisitos do negócio e do produto mudam frequentemente à medida que o desenvolvimento prossegue. Prazos reduzidos impossibilitam criar uma versão completa e abrangente do software, mas é desejável a produção de uma versão reduzida para fazer face às pressões do negócio. Há um conjunto básico de requisitos bem determinado, mas os detalhes e extensões do produto precisam ser definidos.
- II. Os requisitos iniciais estão bem definidos, mas o escopo global do esforço de desenvolvimento não é linear. Há necessidade de se fornecer rapidamente um conjunto limitado de funcionalidades do software aos usuários e depois refinar e expandir aquelas funcionalidades em versões subsequentes do sistema.

Os modelos corretos propostos em I e II, são, respectivamente,

- a) Espiral, que inclui o processo Baseado em Componentes e o Ágil, que inclui o Processo Unificado.
- b) Evolucionário, que inclui os processos Espiral e Concorrente e o Incremental, que inclui o processo RAD.
- c) Prototipagem, que inclui o Processo Unificado e o Incremental, que inclui o processo Concorrente.
- d) Incremental, que inclui o processo RAD e o Sequencial, que inclui o processo em Cascata.
- e) Baseado em Componentes, que inclui a Prototipagem e o Evolucionário, que inclui o Processo Unificado.

Q13 – FCC - 2015 – TRE/RR

Deve-se propor um modelo e processo(s) de software adequado(s) para o desenvolvimento de sistemas com as seguintes características:

- I. Os requisitos do negócio e do produto mudam frequentemente à medida que o desenvolvimento prossegue. Prazos reduzidos impossibilitam criar uma versão completa e abrangente do software, mas é desejável a produção de uma versão reduzida para fazer face às pressões do negócio. Há um conjunto básico de requisitos bem determinado, mas os detalhes e extensões do produto precisam ser definidos.
- II. Os requisitos iniciais estão bem definidos, mas o escopo global do esforço de desenvolvimento não é linear. Há necessidade de se fornecer rapidamente um conjunto limitado de funcionalidades do software aos usuários e depois refinar e expandir aquelas funcionalidades em versões subsequentes do sistema.

Os modelos corretos propostos em I e II, são, respectivamente,

- a) Espiral, que inclui o processo Baseado em Componentes e o Ágil, que inclui o Processo Unificado.
- ➔ b) Evolucionário, que inclui os processos Espiral e Concorrente e o Incremental, que inclui o processo RAD.
- c) Prototipagem, que inclui o Processo Unificado e o Incremental, que inclui o processo Concorrente.
- d) Incremental, que inclui o processo RAD e o Sequencial, que inclui o processo em Cascata.
- e) Baseado em Componentes, que inclui a Prototipagem e o Evolucionário, que inclui o Processo Unificado.

Q14 – FCC - 2014 – TRT2

O modelo de ciclo de vida incremental e iterativo foi proposto como uma resposta aos problemas encontrados no modelo em cascata. Em relação a este tipo de modelo de processo, é INCORRETO afirmar que

- a) um processo de desenvolvimento, segundo essa abordagem, divide o desenvolvimento de um produto de software em ciclos. Em cada ciclo de desenvolvimento podem ser identificadas as fases de análise, projeto, implementação e testes.
- b) cada um dos ciclos de desenvolvimento considera um subconjunto de requisitos. Os requisitos são desenvolvidos uma vez que sejam alocados a um ciclo de desenvolvimento. No próximo ciclo, um outro subconjunto dos requisitos é considerado para ser desenvolvido, o que produz um novo incremento do sistema que contém extensões e refinamentos sobre o incremento anterior.
- c) o desenvolvimento evolui em versões, através da construção incremental e iterativa de novas funcionalidades até que o sistema completo esteja construído. Apenas uma parte dos requisitos é considerada em cada ciclo de desenvolvimento.
- d) o desenvolvimento incremental e iterativo de software sempre é possível, desde que a equipe de desenvolvimento divida os requisitos do sistema em partes, para que cada parte seja alocada a um ciclo de desenvolvimento. Para isso, a Norma NBR ISO/IEC 12207:2009 traz recomendações em sua seção Processos de Contexto de Sistema.
- e) os dois padrões mais conhecidos de sistemas iterativos de desenvolvimento são: o UP (Processo Unificado) e o desenvolvimento ágil de software. Por isso, o desenvolvimento iterativo e incremental é também uma parte essencial do XP.

Q14 – FCC - 2014 – TRT2

O modelo de ciclo de vida incremental e iterativo foi proposto como uma resposta aos problemas encontrados no modelo em cascata. Em relação a este tipo de modelo de processo, é INCORRETO afirmar que

- a) um processo de desenvolvimento, segundo essa abordagem, divide o desenvolvimento de um produto de software em ciclos. Em cada ciclo de desenvolvimento podem ser identificadas as fases de análise, projeto, implementação e testes.
- b) cada um dos ciclos de desenvolvimento considera um subconjunto de requisitos. Os requisitos são desenvolvidos uma vez que sejam alocados a um ciclo de desenvolvimento. No próximo ciclo, um outro subconjunto dos requisitos é considerado para ser desenvolvido, o que produz um novo incremento do sistema que contém extensões e refinamentos sobre o incremento anterior.
- c) o desenvolvimento evolui em versões, através da construção incremental e iterativa de novas funcionalidades até que o sistema completo esteja construído. Apenas uma parte dos requisitos é considerada em cada ciclo de desenvolvimento.
- ➔ d) o desenvolvimento incremental e iterativo de software sempre é possível, desde que a equipe de desenvolvimento divida os requisitos do sistema em partes, para que cada parte seja alocada a um ciclo de desenvolvimento. Para isso, a Norma NBR ISO/IEC 12207:2009 traz recomendações em sua seção Processos de Contexto de Sistema.
- e) os dois padrões mais conhecidos de sistemas iterativos de desenvolvimento são: o UP (Processo Unificado) e o desenvolvimento ágil de software. Por isso, o desenvolvimento iterativo e incremental é também uma parte essencial do XP.

A escolha de um modelo é fortemente dependente das características do projeto. Os principais modelos de ciclo de vida podem ser agrupados em três categorias principais:

- a) sequenciais, cascata e evolutivos.
- b) sequenciais, incrementais e ágeis.
- c) sequenciais, incrementais e evolutivos.
- d) sequenciais, ágeis e cascata
- e) cascata, ágeis e evolutivos.

A escolha de um modelo é fortemente dependente das características do projeto. Os principais modelos de ciclo de vida podem ser agrupados em três categorias principais:

- a) sequenciais, cascata e evolutivos.
- b) sequenciais, incrementais e ágeis.
- ➔ c) sequenciais, incrementais e evolutivos.
- d) sequenciais, ágeis e cascata
- e) cascata, ágeis e evolutivos.

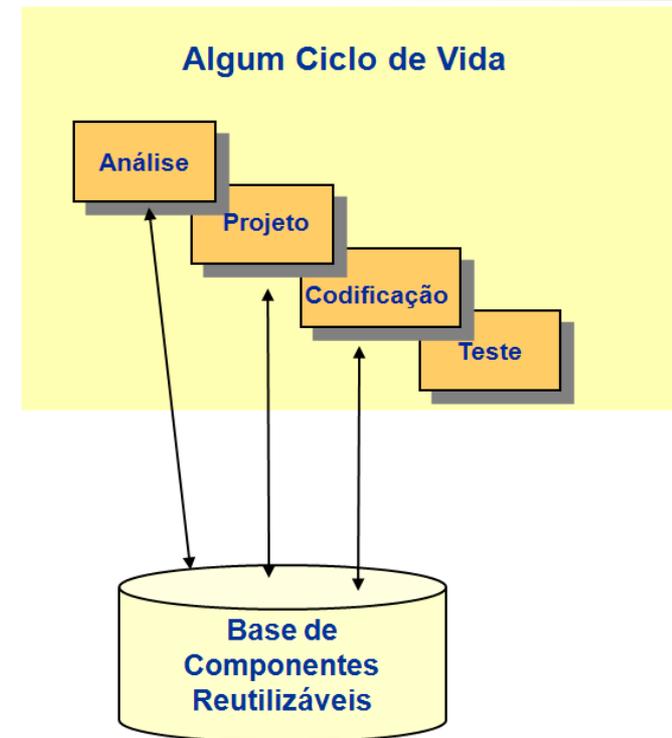
Modelo Especializados

- Levam em conta muita das características dos modelos já vistos
- Abordagem específicas
- Tipos:
 - Desenvolvimento Baseado em Componentes
 - Métodos Formais
 - Orientado a Aspectos

Desenvolvimento Baseado em Componentes

- Enfatiza a reutilização de componentes
- Evolucionário
- Utiliza fortemente UML

- Vantagens:
 - Aumento da produtividade
 - Diminuição do ciclo de desenvolvimento
 - Aumento da qualidade



Métodos Formais

- Baseados em técnicas matemáticas
- Após a “prova” da especificação é transformado em código
- Promessa de um software “livre de defeitos”
- Aplicado a sistemas críticos
- “Notação Z ou B”, Álgebra de Processos em notação LOTOS, Teoria de Conjuntos;
- Uma variante dessa abordagem é denominada Cleanroom

- Vantagens:
 - Precisão e prova matemática
- Desvantagens:
 - Dificuldade de se discutir o modelo com o usuário
 - Tempo necessário para o desenvolvimento
 - Treinamento extensivo

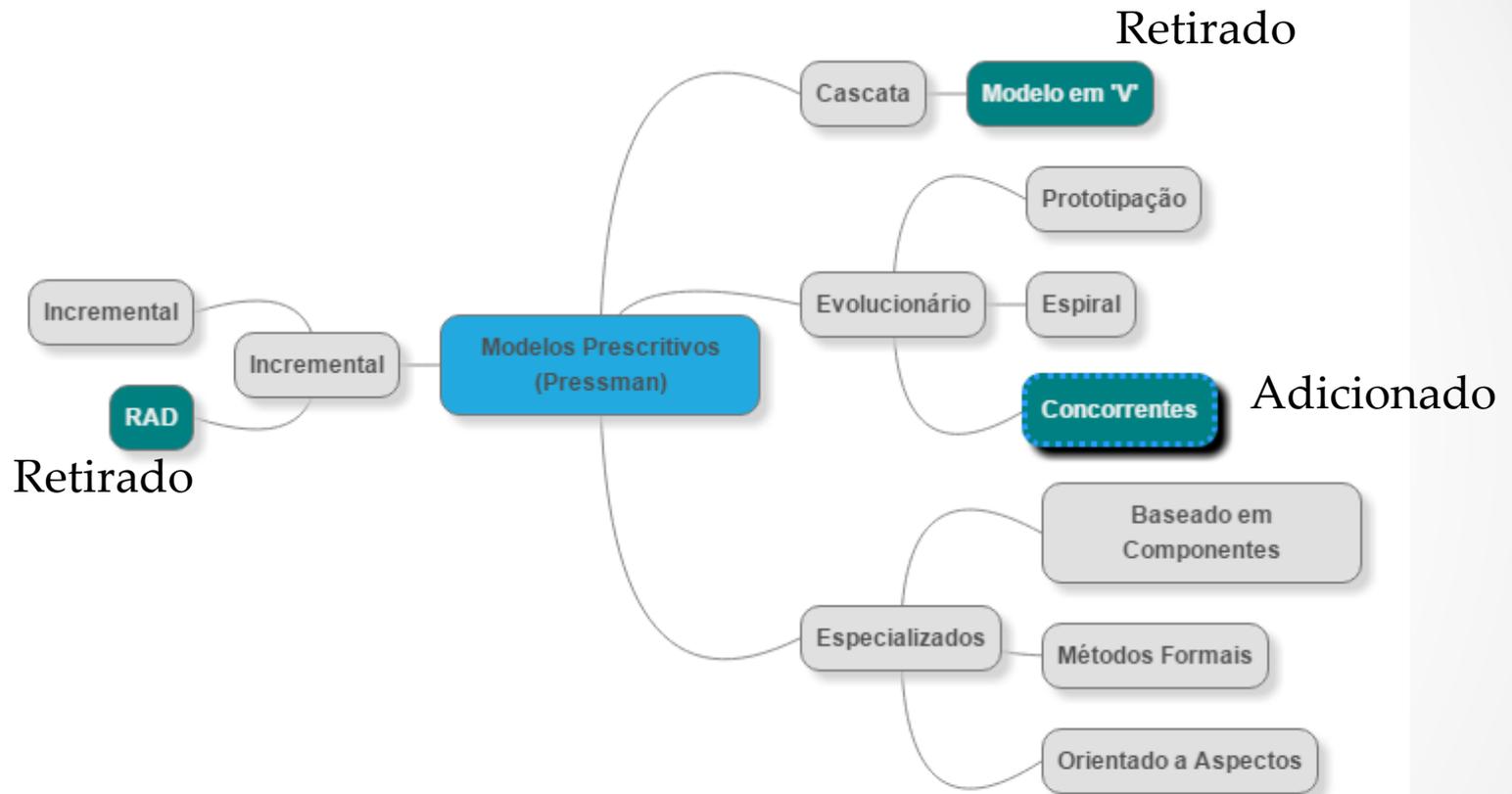
Métodos Formais

- Notação Z
 - Baseada em teoria de conjuntos
- Notação B
 - Baseada em teoria de conjuntos e lógica de predicados
- CSP (Communicating Sequential Processes)
 - Baseada em Álgebra de Processos
- Notação LOTOS (Language of Temporal Ordering Specification)
 - Baseada em Álgebra de Processos
- Redes de Petri
 - Baseada em várias representações matemáticas

Orientado a Aspectos

- Características que não influenciam apenas uma parte de um componente, mas o sistema como um todo
- Alternativa para resolver problemas que não são solucionados pelos demais métodos de desenvolvimento
- Exemplo de características “transversais”:
 - Persistência
 - Auditoria
 - Segurança
 - Log

Sexta x Sétima Edição



Outros Modelos

- Cleanroom
 - Remove a necessidade de depuração do programa assegurando que erros nunca começam introduzidos no sistema.
 - Objetiva garantir que o projeto implementa corretamente a especificação através da revisão por par.
- Code and Fix
 - Construir através de um entendimento preliminar.
 - Implementa a primeira versão.
 - Faz testes e corrige os erros

Q16 – FCC - 2015 – TRE/RR

Deve-se propor um modelo e processo(s) de software adequado(s) para o desenvolvimento de sistemas com as seguintes características:

- I. Os requisitos do negócio e do produto mudam frequentemente à medida que o desenvolvimento prossegue. Prazos reduzidos impossibilitam criar uma versão completa e abrangente do software, mas é desejável a produção de uma versão reduzida para fazer face às pressões do negócio. Há um conjunto básico de requisitos bem determinado, mas os detalhes e extensões do produto precisam ser definidos.
- II. Os requisitos iniciais estão bem definidos, mas o escopo global do esforço de desenvolvimento não é linear. Há necessidade de se fornecer rapidamente um conjunto limitado de funcionalidades do software aos usuários e depois refinar e expandir aquelas funcionalidades em versões subsequentes do sistema.

Os modelos corretos propostos em I e II, são, respectivamente,

- a) Espiral, que inclui o processo Baseado em Componentes e o Ágil, que inclui o Processo Unificado.
- b) Evolucionário, que inclui os processos Espiral e Concorrente e o Incremental, que inclui o processo RAD.
- c) Prototipagem, que inclui o Processo Unificado e o Incremental, que inclui o processo Concorrente.
- d) Incremental, que inclui o processo RAD e o Sequencial, que inclui o processo em Cascata.
- e) Baseado em Componentes, que inclui a Prototipagem e o Evolucionário, que inclui o Processo Unificado.

Q16 – FCC - 2015 – TRE/RR

Deve-se propor um modelo e processo(s) de software adequado(s) para o desenvolvimento de sistemas com as seguintes características:

- I. Os requisitos do negócio e do produto mudam frequentemente à medida que o desenvolvimento prossegue. Prazos reduzidos impossibilitam criar uma versão completa e abrangente do software, mas é desejável a produção de uma versão reduzida para fazer face às pressões do negócio. Há um conjunto básico de requisitos bem determinado, mas os detalhes e extensões do produto precisam ser definidos.
- II. Os requisitos iniciais estão bem definidos, mas o escopo global do esforço de desenvolvimento não é linear. Há necessidade de se fornecer rapidamente um conjunto limitado de funcionalidades do software aos usuários e depois refinar e expandir aquelas funcionalidades em versões subsequentes do sistema.

Os modelos corretos propostos em I e II, são, respectivamente,

- a) Espiral, que inclui o processo Baseado em Componentes e o Ágil, que inclui o Processo Unificado.
- ➔ b) Evolucionário, que inclui os processos Espiral e Concorrente e o Incremental, que inclui o processo RAD.
- c) Prototipagem, que inclui o Processo Unificado e o Incremental, que inclui o processo Concorrente.
- d) Incremental, que inclui o processo RAD e o Sequencial, que inclui o processo em Cascata.
- e) Baseado em Componentes, que inclui a Prototipagem e o Evolucionário, que inclui o Processo Unificado.

Métodos Formais é o termo usado para definir o processo de especificação de software fundamentado em linguagens formais.

Certo

Errado

Métodos Formais é o termo usado para definir o processo de especificação de software fundamentado em linguagens formais.

➡ Certo

Errado

Processos de Software

- Conjunto de atividades relacionadas que levam à produção de um produto de software.
- Existem muitos, no entanto 4 atividades são fundamentais para todos:
 - Especificação de Software
 - Definições, restrições e funcionalidades.
 - Projeto e Implementação de Software
 - Produzir o software de acordo com as especificações.
 - Validação de Software
 - Garantir que atende às necessidades dos clientes.
 - Evolução do Software
 - Evoluir para refletir mudanças solicitadas

Um processo de software é um conjunto de atividades relacionadas que levam à produção de um produto de software. Existem muitos processos de software diferentes, mas todos devem incluir quatro atividades fundamentais: especificação, projeto e implementação, validação e

- a) teste
- b) evolução.
- c) prototipação.
- d) entrega.
- e) modelagem.

Um processo de software é um conjunto de atividades relacionadas que levam à produção de um produto de software. Existem muitos processos de software diferentes, mas todos devem incluir quatro atividades fundamentais: especificação, projeto e implementação, validação e

a) teste

➔ b) evolução.

c) prototipação.

d) entrega.

e) modelagem.

Ciclo de Vida do Software

- Definição
 - Modelagem e análise. Inclui estudo de viabilidade.
- Desenvolvimento
 - Objetiva a construção do produto.
- Operação
 - Distribuição e entrega, instalação, utilização e manutenção.
- Retirada
 - Evolução para novas plataformas operacionais ou incorporação de novos requisitos. Grande desafio atual.

De acordo com a engenharia de software, como todo produto industrial, o software possui um ciclo de vida. Cada fase do ciclo de vida possui divisões e subdivisões. Em qual fase avaliamos a necessidade de evolução dos softwares em funcionamento para novas plataformas operacionais ou para a incorporação de novos requisitos?

- a) Fase de operação;
- b) Fase de retirada;
- c) Fase de definição;
- d) Fase de design.
- e) Fase de desenvolvimento;

De acordo com a engenharia de software, como todo produto industrial, o software possui um ciclo de vida. Cada fase do ciclo de vida possui divisões e subdivisões. Em qual fase avaliamos a necessidade de evolução dos softwares em funcionamento para novas plataformas operacionais ou para a incorporação de novos requisitos?

- a) Fase de operação;
- ➔ b) Fase de retirada;
- c) Fase de definição;
- d) Fase de design.
- e) Fase de desenvolvimento;

De acordo com Sommerville, no gerenciamento e modelagem de negócio, as quatro principais atividades do processo, são:

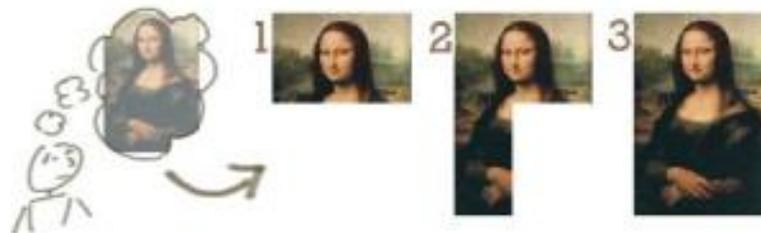
- (A) Estudo, Esboço, Implementação e Teste
- (B) Viabilidade, Planejamento, Projeto e Operação
- (C) Especificação, Desenvolvimento, Validação e Evolução
- (D) Requisito, Anteprojeto, Implementação e Implantação
- (E) Levantamento, Análise, Desenvolvimento e Manutenção

De acordo com Sommerville, no gerenciamento e modelagem de negócio, as quatro principais atividades do processo, são:

- (A) Estudo, Esboço, Implementação e Teste
- (B) Viabilidade, Planejamento, Projeto e Operação
- ➔ (C) Especificação, Desenvolvimento, Validação e Evolução
- (D) Requisito, Anteprojeto, Implementação e Implantação
- (E) Levantamento, Análise, Desenvolvimento e Manutenção

Incremental x Iterativo

Incremental



Iterative



Resumo

- Cascata:
 - Sequencial, linear, clássico
- Incremental:
 - Paralelo, cada equipe alocada em uma tarefa
- RAD:
 - Concorrente, mais de uma equipe em cada incremento
- Prototipação:
 - Descartável: apenas para esclarecer requisitos
 - Evolutiva: protótipo evolui até o sistema real
- Espiral:
 - Riscos, riscos e riscos
- Concorrentes:
 - Estados de cada tarefa
- Componentes:
 - Reutilização
- Métodos Formais:
 - Notação matemática, promessa de qualidade extrema
- Aspectos:
 - Tratar requisitos comuns a todo o sistema

Gabarito

1- B	5- A	9- E	13 – B	17 – CERTO
2- ERRADO	6- A	10- A	14 – D	18 – B
3- CERTO	7- ERRADO	11 – C	15 – C	19 – B
4- CERTO	8- B	12 – B	16 – B	20 - C

Questões Comentadas

Considere as seguintes afirmações feitas sobre um modelo de processo de software.

- I. Combina a natureza iterativa de modelos incrementais com aspectos sistemáticos do modelo em cascata.
- II. Pode ser aplicado em todo ciclo de vida de uma aplicação, inclusive, após a entrega do software.
- III. É um modelo que reconhece explicitamente a necessidade de gerenciar riscos.

As três afirmações se referem ao:

- a) Modelo clássico
- b) Protótipo evolucionário
- c) Processo unificado
- d) Modelo espiral.
- e) Modelo de métodos formais.

Considere as seguintes afirmações feitas sobre um modelo de processo de software.

- I. Combina a natureza iterativa de modelos incrementais com aspectos sistemáticos do modelo em cascata.
- II. Pode ser aplicado em todo ciclo de vida de uma aplicação, inclusive, após a entrega do software.
- III. É um modelo que reconhece explicitamente a necessidade de gerenciar riscos.

As três afirmações se referem ao:

- a) Modelo clássico
- b) Protótipo evolucionário
- c) Processo unificado
- ➔ d) Modelo espiral.
- e) Modelo de métodos formais.

O Ciclo de Vida de um Sistema especifica todas as fases de desenvolvimento, desde sua concepção até o processo de manutenção e declínio. No que diz respeito ao desenvolvimento de software, existem alguns processos conhecidos. Um destes processos, possui característica iterativa e incremental, inicia cada fase do projeto realizando um planejamento prévio, realiza a execução da fase, verifica o progresso e os resultados da fase (riscos, lições aprendidas) e incrementa novos objetivos para a fase seguinte, seguindo para a próxima iteração. O processo de software em questão é o

- a) modelo espiral.
- b) ciclo de vida em cascata.
- c) modelo de desenvolvimento evolucionário (prototipação).
- d) modelo de desenvolvimento ágil.
- e) método de desenvolvimento Cleanroom (Sala Limpa).

O Ciclo de Vida de um Sistema especifica todas as fases de desenvolvimento, desde sua concepção até o processo de manutenção e declínio. No que diz respeito ao desenvolvimento de software, existem alguns processos conhecidos. Um destes processos, possui característica iterativa e incremental, inicia cada fase do projeto realizando um planejamento prévio, realiza a execução da fase, verifica o progresso e os resultados da fase (riscos, lições aprendidas) e incrementa novos objetivos para a fase seguinte, seguindo para a próxima iteração. O processo de software em questão é o

- ➔ a) modelo espiral.
- b) ciclo de vida em cascata.
- c) modelo de desenvolvimento evolucionário (prototipação).
- d) modelo de desenvolvimento ágil.
- e) método de desenvolvimento Cleanroom (Sala Limpa).

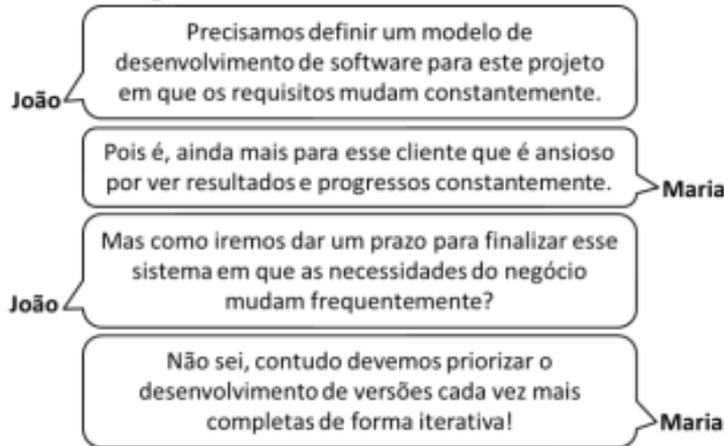
Na Engenharia de Software, há diversos modelos de ciclo de vida, definidos com variados níveis de formalidade. O modelo

- a) cascata (ou clássico) é adequado para controlar riscos e requisitos voláteis durante o desenvolvimento do sistema.
- b) codificação e correção (code and fix) é adequado para alcançar um bom nível de manutenibilidade do sistema.
- c) prototipagem descartável é adequado para descartar a fase de levantamento de requisitos do sistema a ser desenvolvido.
- d) prototipagem evolutiva entrega uma versão inicial do sistema, que considera requisitos já definidos com o cliente.
- e) espiral é inadequado quando são necessários o uso de protótipos durante a validação do sistema e o reúso de software.

Na Engenharia de Software, há diversos modelos de ciclo de vida, definidos com variados níveis de formalidade. O modelo

- a) cascata (ou clássico) é adequado para controlar riscos e requisitos voláteis durante o desenvolvimento do sistema.
- b) codificação e correção (code and fix) é adequado para alcançar um bom nível de manutenibilidade do sistema.
- c) prototipagem descartável é adequado para descartar a fase de levantamento de requisitos do sistema a ser desenvolvido.
- ➔ d) prototipagem evolutiva entrega uma versão inicial do sistema, que considera requisitos já definidos com o cliente.
- e) espiral é inadequado quando são necessários o uso de protótipos durante a validação do sistema e o reúso de software.

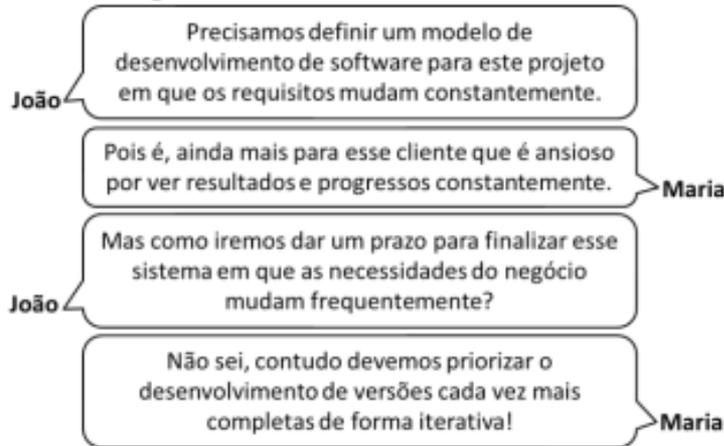
Observe o seguinte diálogo entre dois Analistas de Sistemas:



O modelo de processo de software que atende às características do projeto que João e Maria irão desenvolver é:

- a) Cascata;
- b) Evolucionário;
- c) Codificação e correção;
- d) Incremental;
- e) Modelo em V.

Observe o seguinte diálogo entre dois Analistas de Sistemas:



O modelo de processo de software que atende às características do projeto que João e Maria irão desenvolver é:

- a) Cascata;
- ➔ b) Evolucionário;
- c) Codificação e correção;
- d) Incremental;
- e) Modelo em V.

O ciclo de vida de um software deve apresentar, no mínimo, as etapas de elaboração do documento de visão, levantamento de regras de negócio, especificação/validação de requisitos e prototipagem.

Certo

Errado

O ciclo de vida de um software deve apresentar, no mínimo, as etapas de elaboração do documento de visão, levantamento de regras de negócio, especificação/validação de requisitos e prototipagem.

Certo

➡ Errado

As atividades fundamentais relacionadas ao processo de construção de um *software* incluem a especificação, o desenvolvimento, a validação e a evolução do *software*.

Certo

Errado

As atividades fundamentais relacionadas ao processo de construção de um *software* incluem a especificação, o desenvolvimento, a validação e a evolução do *software*.

→ Certo

Errado

No ciclo de vida da primeira versão do modelo em espiral, a etapa de análise de riscos é realizada dentro da fase de desenvolvimento.

Certo

Errado

No ciclo de vida da primeira versão do modelo em espiral, a etapa de análise de riscos é realizada dentro da fase de desenvolvimento.

Certo

→ Errado

O ciclo de vida de um *software*, entre outras características, está relacionado aos estágios de concepção, projeto, criação e implementação.

Certo

Errado

O ciclo de vida de um *software*, entre outras características, está relacionado aos estágios de concepção, projeto, criação e implementação.

➡ Certo

Errado

No ciclo de vida de *software*, a estrutura de dados, a arquitetura, os detalhes procedimentais e a caracterização da interface são atributos da etapa de análise e engenharia de *software*.

Certo

Errado

No ciclo de vida de *software*, a estrutura de dados, a arquitetura, os detalhes procedimentais e a caracterização da interface são atributos da etapa de análise e engenharia de *software*.

Certo

➡ Errado

- O Ciclo de Vida de um Sistema especifica todas as fases de desenvolvimento, desde sua concepção até o processo de manutenção e declínio. No que diz respeito ao desenvolvimento de software, existem alguns processos conhecidos. Um destes processos, possui característica iterativa e incremental, inicia cada fase do projeto realizando um planejamento prévio, realiza a execução da fase, verifica o progresso e os resultados da fase (riscos, lições aprendidas) e incrementa novos objetivos para a fase seguinte, seguindo para a próxima iteração. O processo de software em questão é o
- a) modelo espiral.
 - b) ciclo de vida em cascata.
 - c) modelo de desenvolvimento evolucionário (prototipação).
 - d) modelo de desenvolvimento ágil.
 - e) método de desenvolvimento Cleanroom (Sala Limpa).

O Ciclo de Vida de um Sistema especifica todas as fases de desenvolvimento, desde sua concepção até o processo de manutenção e declínio. No que diz respeito ao desenvolvimento de software, existem alguns processos conhecidos. Um destes processos, possui característica iterativa e incremental, inicia cada fase do projeto realizando um planejamento prévio, realiza a execução da fase, verifica o progresso e os resultados da fase (riscos, lições aprendidas) e incrementa novos objetivos para a fase seguinte, seguindo para a próxima iteração. O processo de software em questão é o

- ➔ a) modelo espiral.
- b) ciclo de vida em cascata.
- c) modelo de desenvolvimento evolucionário (prototipação).
- d) modelo de desenvolvimento ágil.
- e) método de desenvolvimento Cleanroom (Sala Limpa).

Considere: é uma disciplina que se ocupa de todos os aspectos da produção de software, desde os estágios iniciais de especificação do sistema até a manutenção desse sistema, depois que ele entrou em operação. Seu principal objetivo é fornecer uma estrutura metodológica para a construção de software com alta qualidade. A definição refere-se

- a) ao ciclo de vida do software.
- b) à programação orientada a objetos.
- c) à análise de sistemas.
- d) à engenharia de requisitos.
- e) à engenharia de software.

Considere: é uma disciplina que se ocupa de todos os aspectos da produção de software, desde os estágios iniciais de especificação do sistema até a manutenção desse sistema, depois que ele entrou em operação. Seu principal objetivo é fornecer uma estrutura metodológica para a construção de software com alta qualidade. A definição refere-se

- a) ao ciclo de vida do software.
- b) à programação orientada a objetos.
- c) à análise de sistemas.
- d) à engenharia de requisitos.
- ➔ e) à engenharia de software.

Em relação aos modelos de processos de software, pode-se dizer que os modelos incremental e evolucionário possuem a característica de serem iterativos. Assinale a alternativa que melhor descreve um modelo de produção de software iterativo.

- a) Os incrementos de um software são entregues ao cliente de uma só vez.
- b) Um modelo de produção de software iterativo é composto pelas fases de análise de requisitos, projeto, implementação, testes (validação), integração e manutenção de software.
- c) A abordagem iterativa possibilita desenvolver um sistema de software de forma incremental, permitindo ao desenvolvedor tirar vantagem daquilo que foi aprendido durante a fase inicial de desenvolvimento de uma versão do sistema. O aprendizado ocorre simultaneamente tanto para o desenvolvedor, quanto para o usuário do sistema.
- d) Permite que sejam desenvolvidas versões cada vez mais completas do software.

Em relação aos modelos de processos de software, pode-se dizer que os modelos incremental e evolucionário possuem a característica de serem iterativos. Assinale a alternativa que melhor descreve um modelo de produção de software iterativo.

- a) Os incrementos de um software são entregues ao cliente de uma só vez.
- b) Um modelo de produção de software iterativo é composto pelas fases de análise de requisitos, projeto, implementação, testes (validação), integração e manutenção de software.
- ➔ c) A abordagem iterativa possibilita desenvolver um sistema de software de forma incremental, permitindo ao desenvolvedor tirar vantagem daquilo que foi aprendido durante a fase inicial de desenvolvimento de uma versão do sistema. O aprendizado ocorre simultaneamente tanto para o desenvolvedor, quanto para o usuário do sistema.
- d) Permite que sejam desenvolvidas versões cada vez mais completas do software.

Dos diferentes modelos para o ciclo de vida de desenvolvimento de um software é correto afirmar que

- a) o modelo em espiral é o mais simples e o mais antigo.
- b) o modelo em cascata é o menos flexível e mais simples.
- c) a fase de especificação de requisitos pode estar ausente do modelo.
- d) a fase de implementação é sempre a última do modelo.
- e) o modelo em cascata é o mais recente e comple-xo.

Dos diferentes modelos para o ciclo de vida de desenvolvimento de um software é correto afirmar que

- a) o modelo em espiral é o mais simples e o mais antigo.
- ➔ b) o modelo em cascata é o menos flexível e mais simples.
- c) a fase de especificação de requisitos pode estar ausente do modelo.
- d) a fase de implementação é sempre a última do modelo.
- e) o modelo em cascata é o mais recente e comple-xo.

Durante a fase inicial do ciclo de vida do desenvolvimento de sistemas, na etapa de investigação, a tarefa que determina a probabilidade de sucesso do sistema proposto e propicia uma avaliação superficial da área técnica, econômica e comportamental do projeto, sendo decisivamente importante para o processo do desenvolvimento de sistemas é chamada

- a) estudo de caso.
- b) análise de requisitos.
- c) análise de equilíbrio.
- d) estudo de viabilidade.
- e) design lógico do sistema.

Durante a fase inicial do ciclo de vida do desenvolvimento de sistemas, na etapa de investigação, a tarefa que determina a probabilidade de sucesso do sistema proposto e propicia uma avaliação superficial da área técnica, econômica e comportamental do projeto, sendo decisivamente importante para o processo do desenvolvimento de sistemas é chamada

- a) estudo de caso.
- b) análise de requisitos.
- c) análise de equilíbrio.
- ➔ d) estudo de viabilidade.
- e) design lógico do sistema.

O teste de sistemas envolve a integração de dois ou mais componentes que implementam funções ou características e depois o teste do sistema integrado. Em dois processos de desenvolvimento, o primeiro se concentra no teste de um incremento que será entregue ao cliente, e o segundo no teste de todo o sistema. Esses processos de desenvolvimento são, respectivamente, dos seguintes tipos

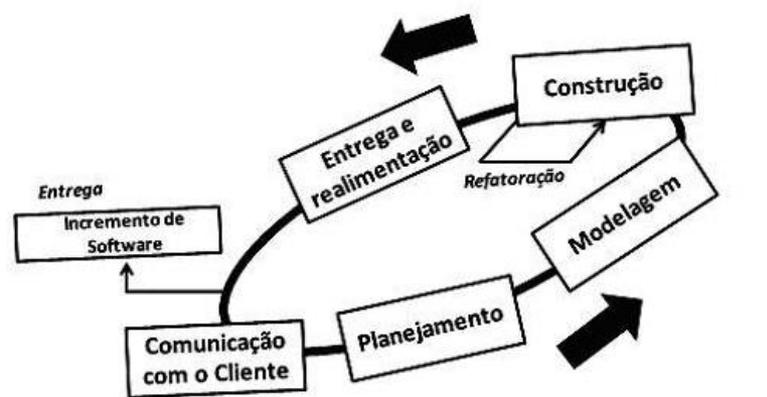
- a) iterativo e em cascata.
- b) hierárquico e em cascata.
- c) iterativo e por prototipação.
- d) hierárquico e por prototipação.

O teste de sistemas envolve a integração de dois ou mais componentes que implementam funções ou características e depois o teste do sistema integrado. Em dois processos de desenvolvimento, o primeiro se concentra no teste de um incremento que será entregue ao cliente, e o segundo no teste de todo o sistema. Esses processos de desenvolvimento são, respectivamente, dos seguintes tipos

- ➔ a) iterativo e em cascata.
- b) hierárquico e em cascata.
- c) iterativo e por prototipação.
- d) hierárquico e por prototipação.

Q36 – CONSULPLAN - 2012 – TSE

No que diz respeito ao desenvolvimento de sistemas para a Web, analise a figura que ilustra um fluxo de processo incremental.



Dentre as diversas atividades desenvolvidas, duas são descritas a seguir.

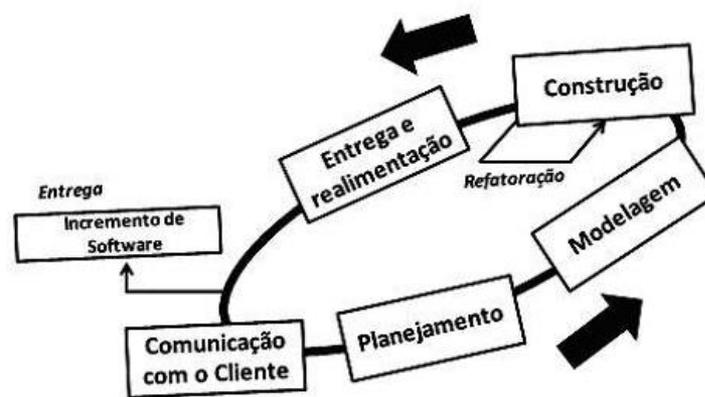
- I. Formulação da análise do negócio.
- II. Detalhamento da arquitetura, navegação e interface.

Estas atividades são executadas, respectivamente, nos seguintes processos

- a) Entrega e realimentação / Construção.
- b) Comunicação com o cliente / Modelagem.

Q36 – CONSULPLAN - 2012 – TSE

No que diz respeito ao desenvolvimento de sistemas para a Web, analise a figura que ilustra um fluxo de processo incremental.



Dentre as diversas atividades desenvolvidas, duas são descritas a seguir.

- I. Formulação da análise do negócio.
- II. Detalhamento da arquitetura, navegação e interface.

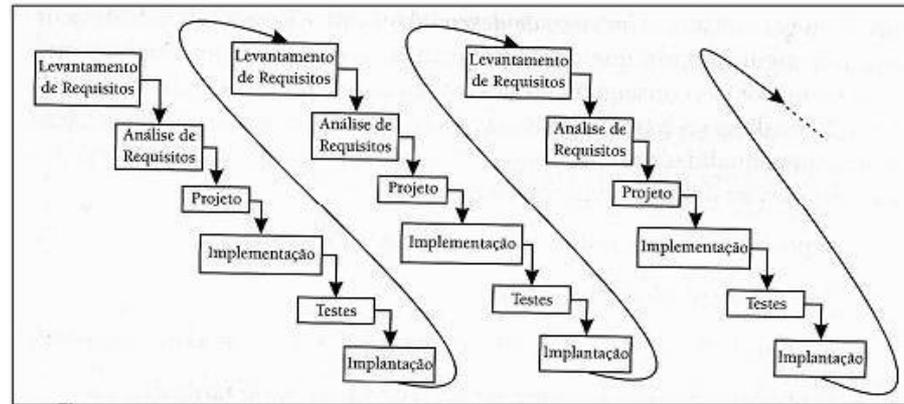
Estas atividades são executadas, respectivamente, nos seguintes processos

a) Entrega e realimentação / Construção.

➡ b) Comunicação com o cliente / Modelagem.

Q37 – CONSULPLAN - 2012 – TSE

Observe um modelo de ciclo de vida para desenvolvimento de sistemas. Nessa abordagem, o desenvolvimento do produto de software é dividido em ciclos, sendo identificadas em cada ciclo, as fases de análise, projeto, implementação e testes.

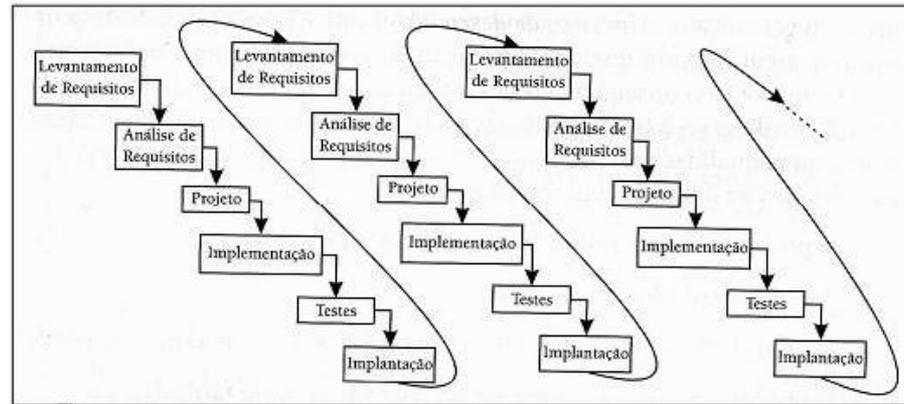


Este modelo é conhecido como ciclo de vida

- a) por prototipação em cascata.
- b) por estágios em módulos.
- c) iterativo e incremental.
- d) evolutivo e procedural.

Q37 – CONSULPLAN - 2012 – TSE

Observe um modelo de ciclo de vida para desenvolvimento de sistemas. Nessa abordagem, o desenvolvimento do produto de software é dividido em ciclos, sendo identificadas em cada ciclo, as fases de análise, projeto, implementação e testes.



Este modelo é conhecido como ciclo de vida

- a) por prototipação em cascata.
- b) por estágios em módulos.
- ➔ c) iterativo e incremental.
- d) evolutivo e procedural.

No desenvolvimento de software em espiral (Boehm), cada loop está dividido em quatro setores. NÃO se trata da denominação de um destes setores:

- a) levantamento.
- b) definição de objetivos.
- c) avaliação e redução de riscos
- d) desenvolvimento e validação.
- e) planejamento.

No desenvolvimento de software em espiral (Boehm), cada loop está dividido em quatro setores. NÃO se trata da denominação de um destes setores:

a) levantamento.

b) definição de objetivos.

➔ c) avaliação e redução de riscos

d) desenvolvimento e validação.

e) planejamento.

A engenharia de *software* engloba processos, métodos e ferramentas. Um de seus focos é a produção de *software* de alta qualidade a custos adequados.

Certo

Errado

A engenharia de *software* engloba processos, métodos e ferramentas. Um de seus focos é a produção de *software* de alta qualidade a custos adequados.

 Certo

Errado

A engenharia de *software*, disciplina relacionada aos aspectos da produção de *software*, abrange somente os processos técnicos do desenvolvimento de *software*.

Certo

Errado

A engenharia de *software*, disciplina relacionada aos aspectos da produção de *software*, abrange somente os processos técnicos do desenvolvimento de *software*.

Certo

→ Errado

Para a utilização de metodologias modernas, com abordagem da engenharia de *software*, recomenda-se a elaboração dos manuais do sistema ao final do projeto, quando todos os seus detalhes já estão definidos.

Certo

Errado

Para a utilização de metodologias modernas, com abordagem da engenharia de *software*, recomenda-se a elaboração dos manuais do sistema ao final do projeto, quando todos os seus detalhes já estão definidos.

Certo

→ Errado

No processo de desenvolvimento de software, o modelo de ciclo de vida em que um sistema de software é desenvolvido em vários passos similares e, em cada passo, o sistema é estendido com mais funcionalidades é denominado de modelo

- A) incremental e iterativo.
- B) concorrente.
- C) de prototipação.
- D) em cascata.
- E) de desenvolvimento ágil.

No processo de desenvolvimento de software, o modelo de ciclo de vida em que um sistema de software é desenvolvido em vários passos similares e, em cada passo, o sistema é estendido com mais funcionalidades é denominado de modelo

- ➔ A) incremental e iterativo.
- B) concorrente.
- C) de prototipação.
- D) em cascata.
- E) de desenvolvimento ágil.

Gabarito Questões Comentadas

21- D	27- ERRADO	33- B	39 – CERTO
22 – A	28- CERTO	34- D	40- ERRADO
23 – D	29- ERRADO	35- A	41 – ERRADO
24 – B	30- A	36 – B	42 - A
25 - ERRADO	31- E	37 – C	
26 – CERTO	32- C	38 – C	