Engenharia de Software

Metodologias Tradicionais de Desenvolvimento de Software

Marcio Victorino

Sumário

- Engenharia de Software
  - Conceitos
- Modelos de Processo de Software
  - Modelo sequencial Linear
  - Modelos Incrementais
  - Modelos Evolucionários
  - Modelos Especializados
  - Engenharia Dirigida a Modelos (MDE)

Conceitos

Engenharia de Software

- É uma disciplina da engenharia 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 produção. Sua meta é o desenvolvimento de sistemas de Sw com boa relação custo-benefício.

Engenharia de Software

- Engenharia de Software X Ciência da Computação
  - A ciência da computação se preocupa com as teorias e os métodos básicos referentes aos computadores, enquanto a engenharia de software se dedica aos problemas práticos da produção de software.

Engenharia de Software

- Engenharia de Software X Engenharia de Sistemas
  - A engenharia de Sistemas com base em computadores se ocupa de todos os aspectos do desenvolvimento e evolução da sistemas complexos, em que o software desempenha papel principal. Ela se ocupa com:
    - Desenvolvimento de hardware.
    - Projeto de políticas e processos.
    - Implantação de sistemas.
    - Engenharia de Software.

Obs: Um sistema é uma coleção significativa de componentes inter-relacionados, que trabalham em conjunto para atingir algum objetivo.

Engenharia de Software

- Evolução:
  - Iniciou em 1968: "crise do software".
  - Década de 1970: Programação Estruturada e Projeto Estruturado.
  - Década de 1980: Análise Estruturada (surgimento de ferramentas CASE).
  - Década de 1990: Análise e Projeto Orientados a Objetos.
    - Orientação a objetos (OO) é um termo geral que inclui qualquer estilo de desenvolvimento de sistemas que seja baseado no conceito de 'objeto', uma entidade que exibe características e comportamentos. A estratégia OO pode ser aplicada à programação e a análise e projeto de sistemas.
  - Década de 2000: SOA.

Engenharia de Software

- Seu objetivo é produzir software:
  - de alta qualidade;
  - que atenda às necessidades dos usuários;
  - dentro de um cronograma e de um orçamento previsíveis.

Camadas da Engenharia de Software

- Qualidade: é a pedra fundamental que sustenta a engenharia de software.
- Processo: é a base para a engenharia de software. É a liga que mantém as camadas de tecnologia coesas e possibilita o desenvolvimento de software de forma racional e dentro do prazo.
- Métodos: fornecem as informações técnicas para desenvolver software. Baseiam-se em um conjunto de princípios básicos que governam cada área da tecnologia e inclui atividades de modelagem e outras técnicas descritivas.
- Ferramentas: fornecem suporte automatizado ou semiautomatizado para o processo e para os métodos.

Engenharia de Software

- Software: são os programas de computador e a documentação associada.
- Processo de Software: é um conjunto de atividades, cuja meta é o desenvolvimento ou a evolução do software.
- Modelo de Processo de Software: é uma representação simplificada de um processo de software, apresentada a partir de uma perspectiva específica.
- Métodos de Engenharia de Software: são abordagens estruturadas para o desenvolvimento de software, que incluem modelos de sistemas, notações, regras, recomendações de projetos e diretrizes de processos.

Engenharia de Software

- Atributos de um bom software:
  - proporcionar funcionalidades e desempenho requeridos;
  - passível de manutenção, confiável e de fácil uso.
- Desafios da Engenharia de SW:
  - lidar com sistemas legado;
  - atender à crescente diversidade;
  - prazos reduzidos.
- Custos da Engenharia de SW:
  - desenvolvimento ≡ 60%
  - testes ≡ 40%

Engenharia de Software

- Essência:
  - **Entenda o problema:** levantamento de requisitos e análise.
  - **Planeje uma solução:** projeto.
  - **Execute o plano:** implementação.
  - **Examine o resultado quanto a precisão:** teste e garantia de qualidade.

Engenharia de Software

- Princípios:
  - **Razão por que tudo existe:** para fornecer valor aos seus usuários.
  - **Mantenha a coisa simples:** todo projeto deve ser tão simples quanto possível, mas não mais simples.
  - **Mantenha a visão:** uma visão clara é essencial para o sucesso de um projeto de software.
  - **O que você produz os outros vão consumir:** sempre especifique, projete e implemente sabendo que mais alguém terá de entender o que você está fazendo.
  - **Esteja aberto para o futuro:** nunca projete a si mesmo em um beco sem saída.
  - **Planeje com antecedência o reuso:** reduz custo e aumenta o valor dos componentes e do sistema ao qual são incorporados.
  - **Pense:** raciocinar clara e completamente antes da ação quase sempre produz melhores resultados.

Processos de Software

- É um conjunto de atividades e resultados associados que levam à produção de um produto de software.
- Não há um processo ideal. Diferentes organizações desenvolveram abordagens diferentes para o desenvolvimento de software.
- Atividades fundamentais:
  - Especificação do software.
  - Projeto e implementação do software.
  - Validação do software.
  - Evolução do software.

[CESPE – SLU – 2019] O processo de desenvolvimento de software, independentemente do seu tamanho e da sua destinação, pode envolver atividades genéricas como comunicação, planejamento, modelagem, construção e uso.

[CESPE – FUNPRESPE – 2016] No gerenciamento do ciclo de vida de um software, o processo de desenvolvimento (processo básico) objetiva estabelecer os requisitos dos componentes e suas interfaces em um sistema.

[CESPE – MPOG – 2015] No ciclo de vida do software, o congelamento dos requisitos do software garante que este, quando em desenvolvimento, atenda à expectativa do usuário, desde que tudo que tenha sido requisitado seja implementado.

[CESPE – STJ – 2015] Embora os engenheiros de software geralmente utilizem uma abordagem sistemática, a abordagem criativa e menos formal pode ser eficiente em algumas circunstâncias, como, por exemplo, para o desenvolvimento de sistemas web, que requerem uma mistura de habilidades de software e de projeto.

[CESPE – STJ – 2015] O foco da engenharia de software inclui especificação do sistema, desenvolvimento de hardware, elaboração do projeto de componentes de hardware e software, definição dos processos e implantação do sistema.

[CESPE – STJ – 2015] As principais atividades de engenharia de software são especificação, desenvolvimento, validação e evolução.

[CESPE – ANTT – 2013] Engenheiros de softwares devem compreender a engenharia de sistemas, que aborda os aspectos de sistemas sociotécnicos.

[CESPE – ANTT – 2013] Produtos de software podem ser construídos utilizando-se diferentes modelos de processo, pois alguns modelos são mais adequados que outros para determinados tipos de aplicação.

[CESPE – TRT/10 – 2013] As atividades fundamentais relacionadas ao processo de construção de um software incluem a especificação, o desenvolvimento, a validação e a evolução do software.

[CESPE – TRT/10 – 2013] A engenharia de software engloba processos, métodos e ferramentas. Um de seus focos é a produção de software de alta qualidade a custos adequados.

[CESPE – TRT/10 – 2013] O ciclo de vida de um software, entre outras características, está relacionado aos estágios de concepção, projeto, criação e implementação.

[CESPE – STF – 2013] Na área de conhecimento ferramentas e métodos, o termo ferramentas se refere à estruturação da atividade de desenvolvimento e manutenção de software com o objetivo de torná-la sistemática; métodos dizem respeito à automação do processo de engenharia de software.

[CESPE – STF – 2013] Engenharia de software é um ramo da ciência da computação que se preocupa com o desenvolvimento de códigos que sejam confiáveis, seguros, baratos e que, quando executados, funcionem e tenham o desempenho desejado.

Modelos de Processo de Software

DIREÇÃO CONCURSOS

CURSOS ONLINE DOMINANDO TI

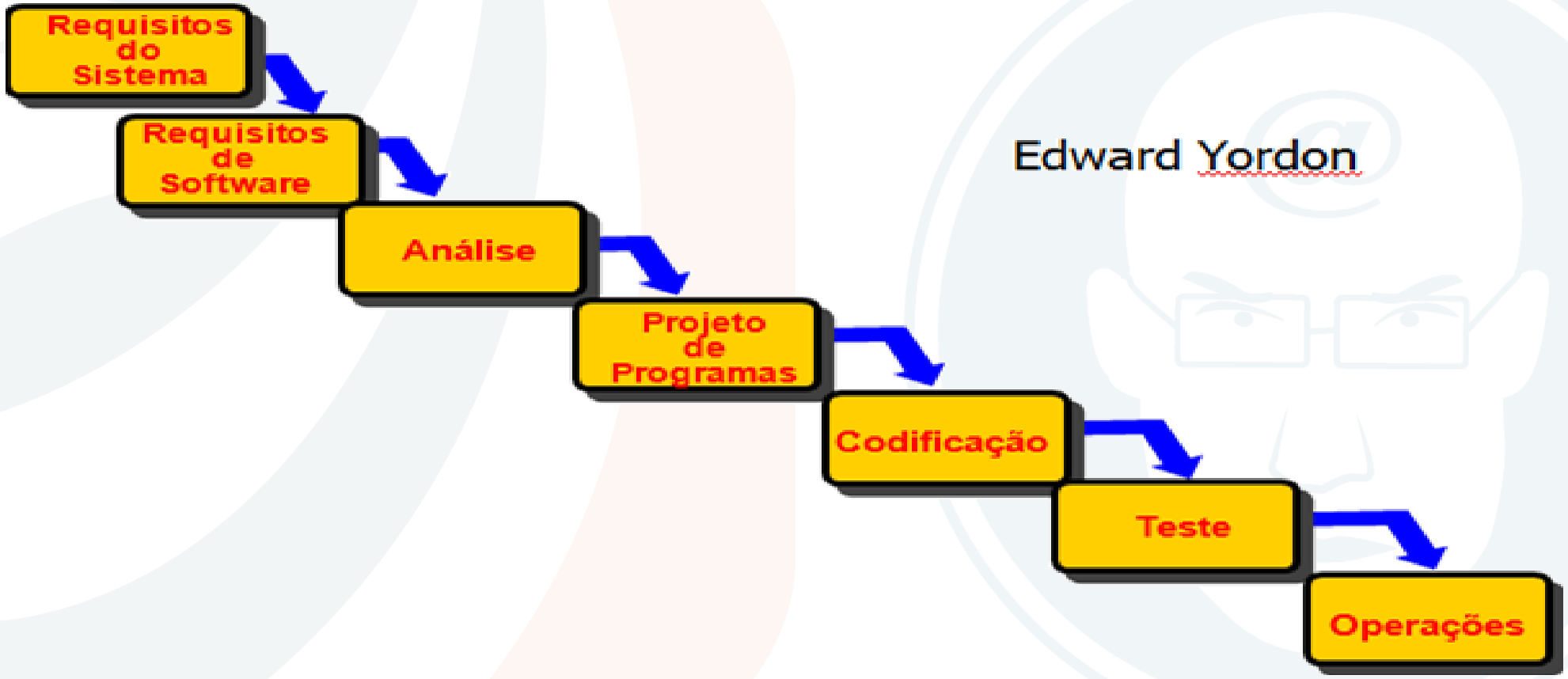
Modelos Tradicionais de Processos de Sw

- Modelo Sequencial Linear.
- Modelos Incrementais.
- Modelos Evolucionários.
- Modelos Especializados.
- Engenharia Dirigida a Modelos (MDE).

Modelo Sequencial Linear

Sequencial Linear

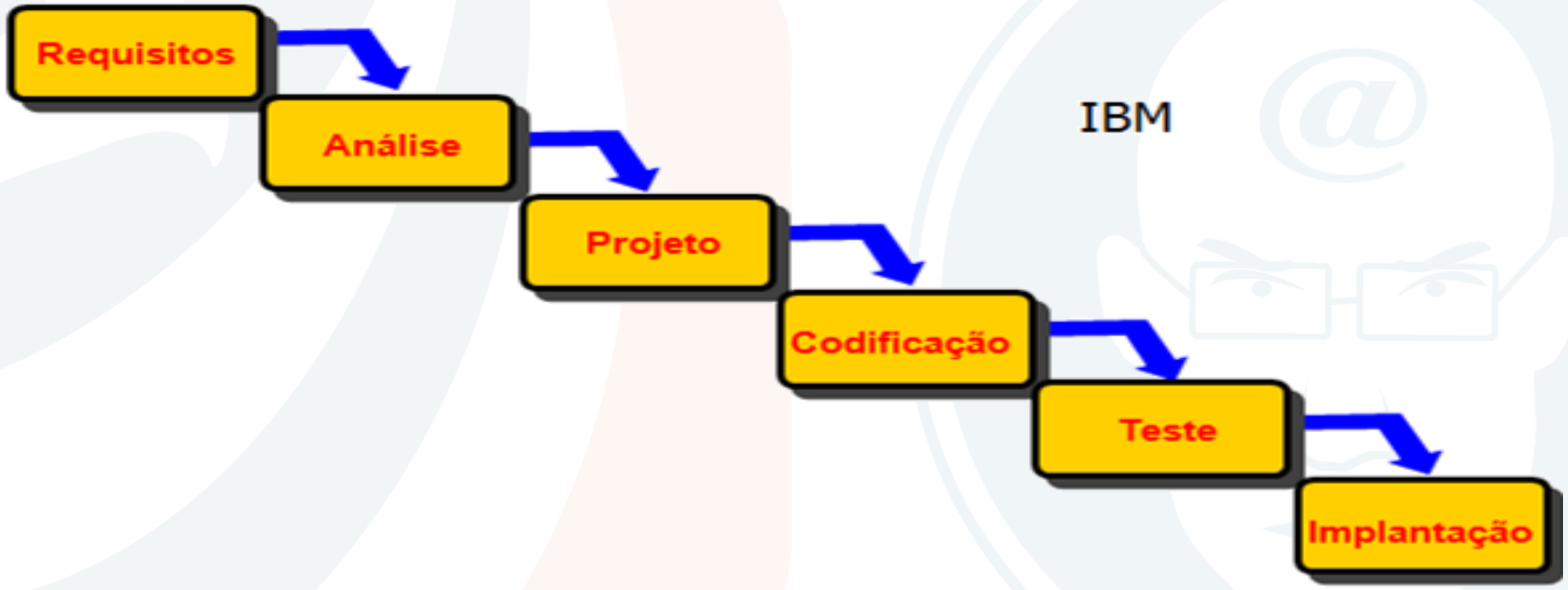
- Modelo Cascata (Clássico):



Edward Yordon

Sequencial Linear

- Modelo Cascata (Clássico):



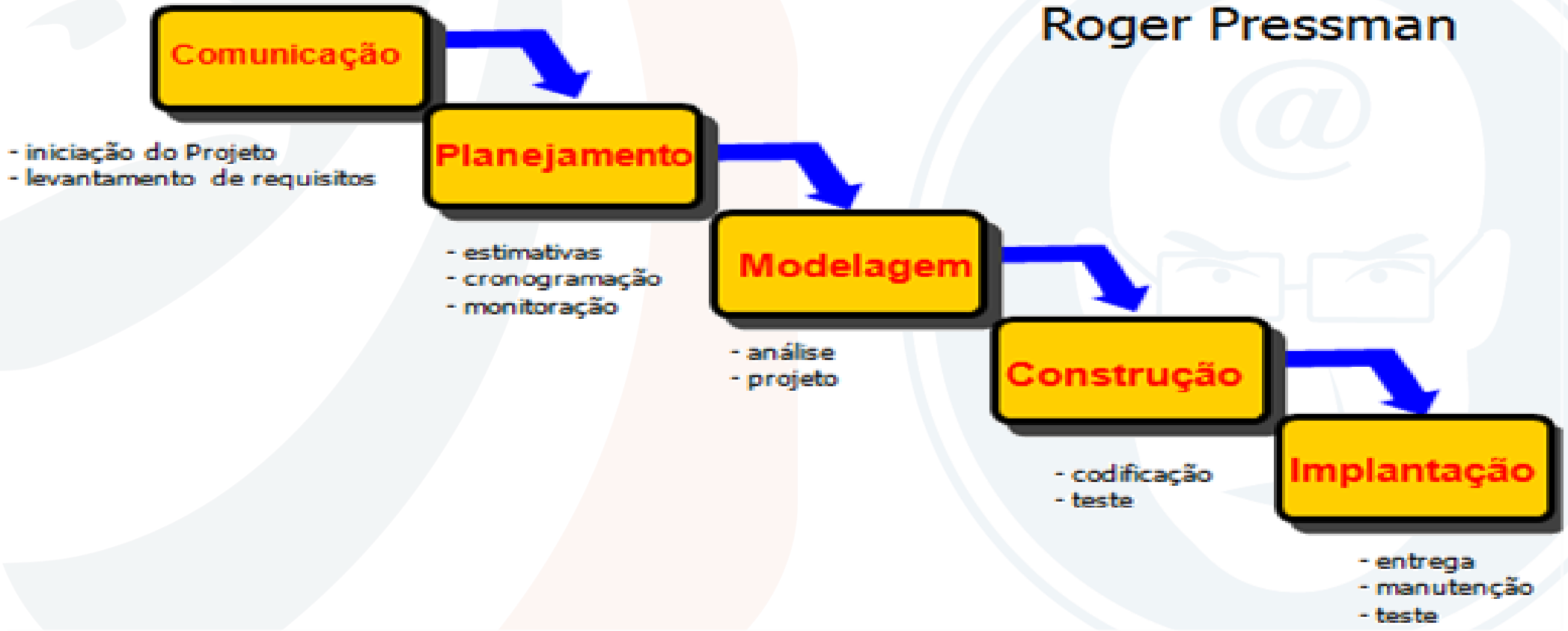
IBM

Modelo Cascata

- **Levantamento de Requisitos:** tem por objetivo propiciar que usuários e desenvolvedores tenham a mesma compreensão do problema a ser resolvido.
- **Análise:** tem por objetivo construir modelos que determinam qual é o problema para o qual estamos tentando conceber uma solução de software.
- **Projeto:** estágio no qual o modelo de análise terá de ser adaptado de tal modo que possa servir como base para implementação no ambiente alvo.
- **Codificação (implementação):** a codificação do sistema é efetivamente executada.
- **Teste:** consiste na verificação do software.
- **Implantação:** entrada em produção do sistema.

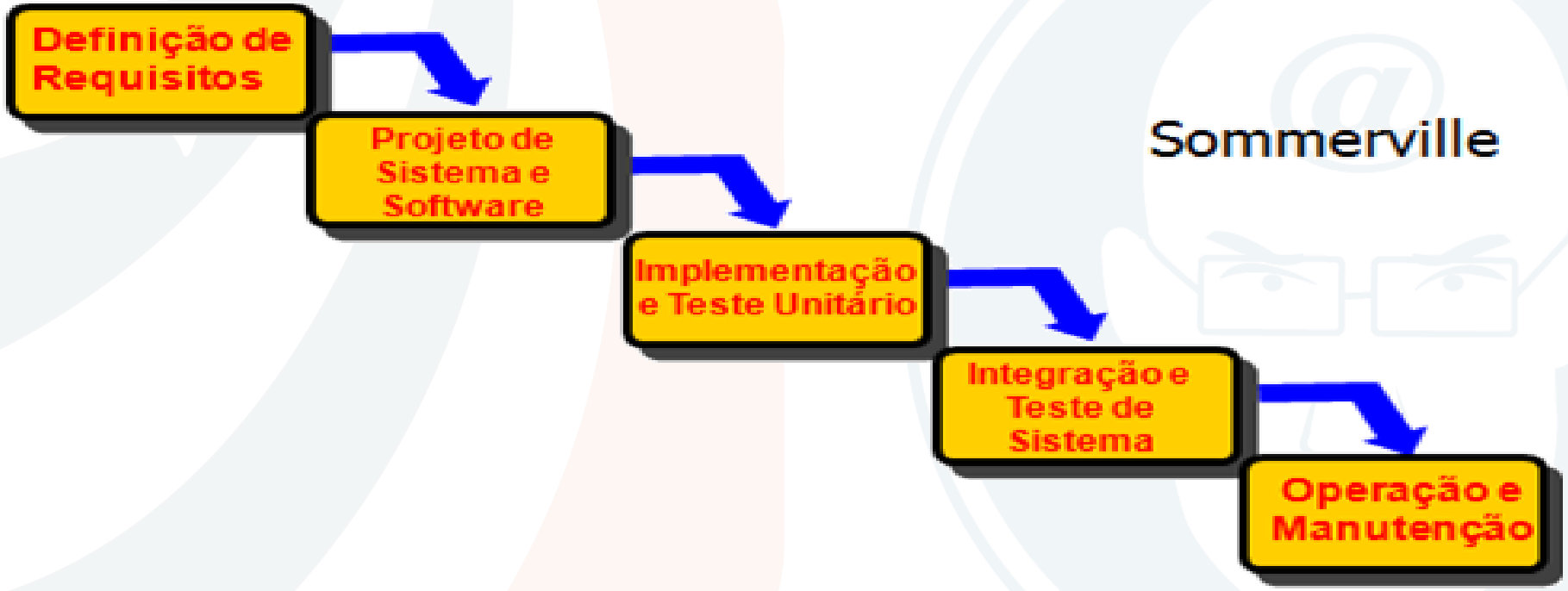
Sequencial Linear

- Modelo Cascata (Clássico):



Sequencial Linear

- Modelo Cascata (Clássico):



Sommerville

Modelo Cascata

- **Análise e definição de requisitos:** Os serviços, restrições e objetivos do sistema são definidos por meio de consulta aos usuários do sistema. Eles são, portanto, definidos detalhadamente e servem como uma especificação de sistema.
- **Projeto de sistema e software:** O processo de projeto de sistema divide os requisitos em sistemas de hardware ou de software. Ele estabelece uma arquitetura geral do sistema. O projeto de software envolve a identificação e a descrição das abstrações fundamentais do sistema de software e suas relações.
- **Implementação e teste de unidade:** Durante esse estágio, o projeto de software é realizado como um conjunto de programas ou unidades de programa. O teste unitário envolve a verificação de que cada unidade atende à sua especificação.
- **Integração e teste de sistema:** As unidades individuais de programa ou os programas são integrados e testados como um sistema completo para garantir que os requisitos de software foram atendidos. Após os testes, o sistema de software é liberado para o cliente.
- **Operação e manutenção:** O sistema é instalado e colocado em operação. A manutenção envolve a correção de erros não detectados nos estágios anteriores do ciclo de vida, no aprimoramento da implementação das unidades de sistema e na ampliação dos serviços de sistema à medida que novos requisitos são identificados.

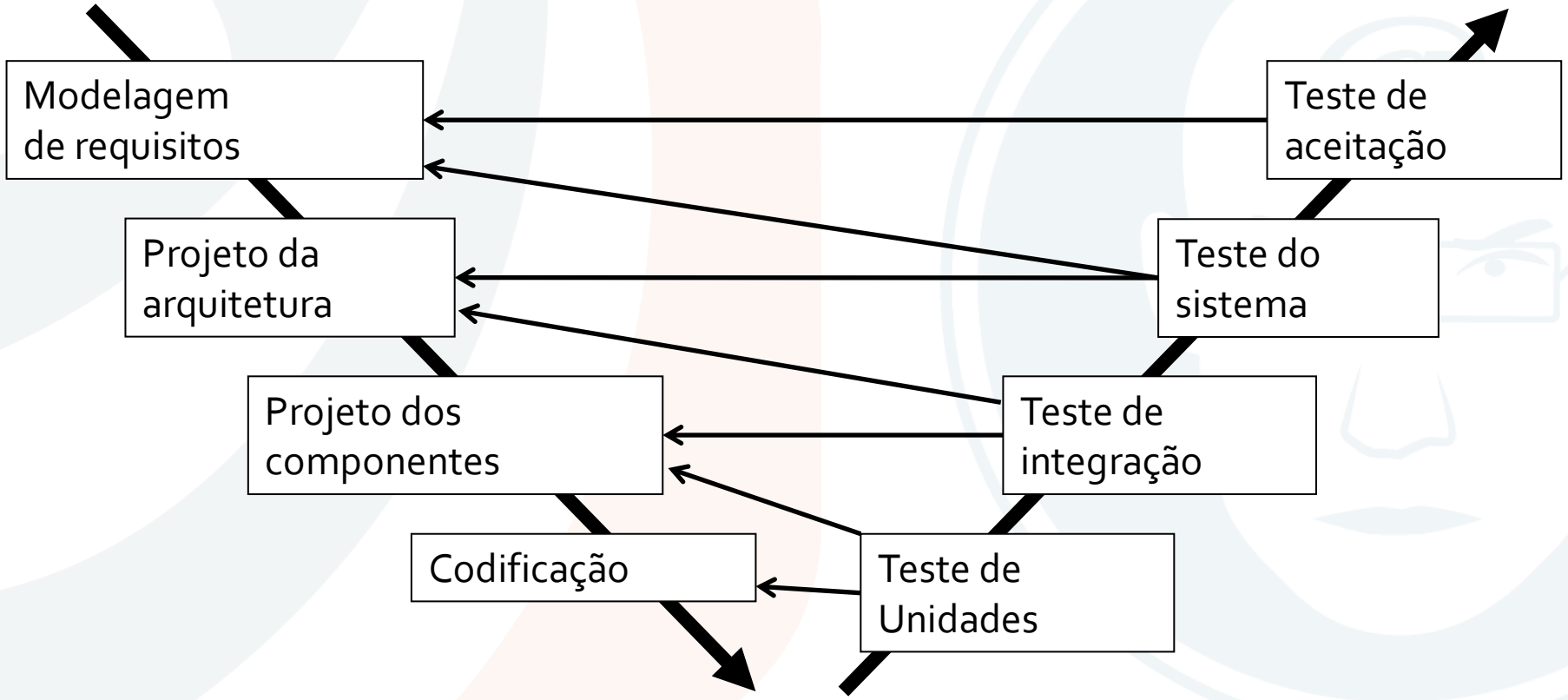
Modelo Cascata

- Problemas:
  - Projetos raramente seguem fluxos sequenciais.
  - Mudanças de requisitos podem causar confusões.
  - É difícil para o cliente estabelecer todas as necessidades no início do projeto.
  - Uma versão dos software só estará disponível próximo ao fim do projeto.
  - Dificulta o gerenciamento de riscos.

Obs: pode ser utilizado quando os requisitos são bem conhecidos e razoavelmente estáveis.

Modelo V (Pressman, p. 60)

- O modelo V é uma variação do Cascata. Este modelo descreve o paralelismo entre as atividades de desenvolvimento e teste de software.



[CESPE – SLU – 2019] No modelo de desenvolvimento de software em cascata, a abordagem é orientada ao risco e as tarefas são organizadas nos seguintes ciclos: determinar objetivos, identificar e resolver riscos, desenvolver e testar, e planejar a próxima iteração.

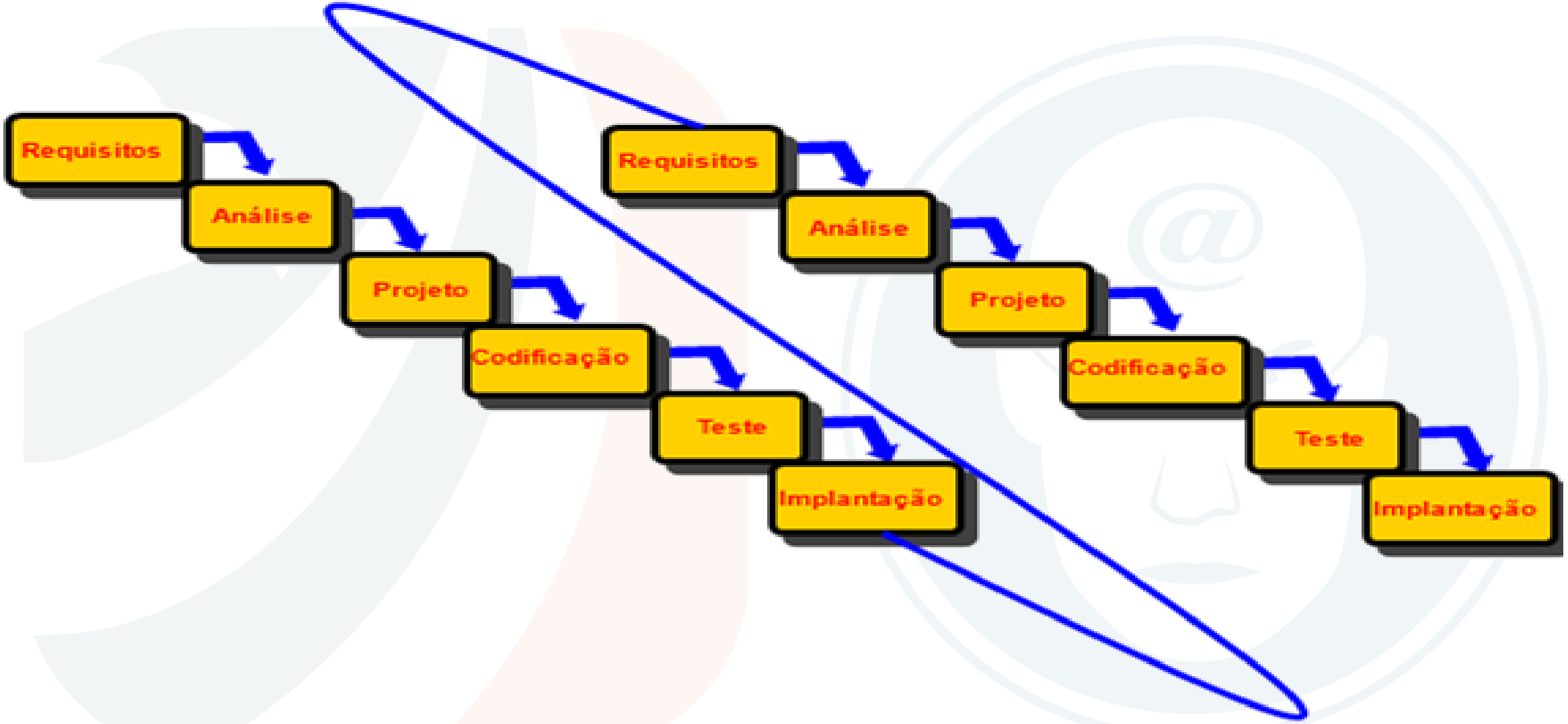
[CESPE – IPHAN – 2018] No estágio de projeto de sistema e software no ciclo de vida de um software, são alocados os requisitos tanto para os sistemas de hardware quanto para os de software, por meio da definição de uma arquitetura geral do sistema.

[CESPE – IPHAN – 2018] No modelo em cascata, com exceção do sequenciamento dos estágios de requisitos e de análise, os demais são executados em paralelo, iniciando-se antes do término do estágio seguinte.

[CESPE – ANATEL – 2014] No ciclo de vida clássico, ou modelo em cascata, as fases concluídas são congeladas para se evitar o retrabalho e maximizarem-se os custos de produção e a aprovação de documentos.

Modelos Incrementais

Modelos Incrementais

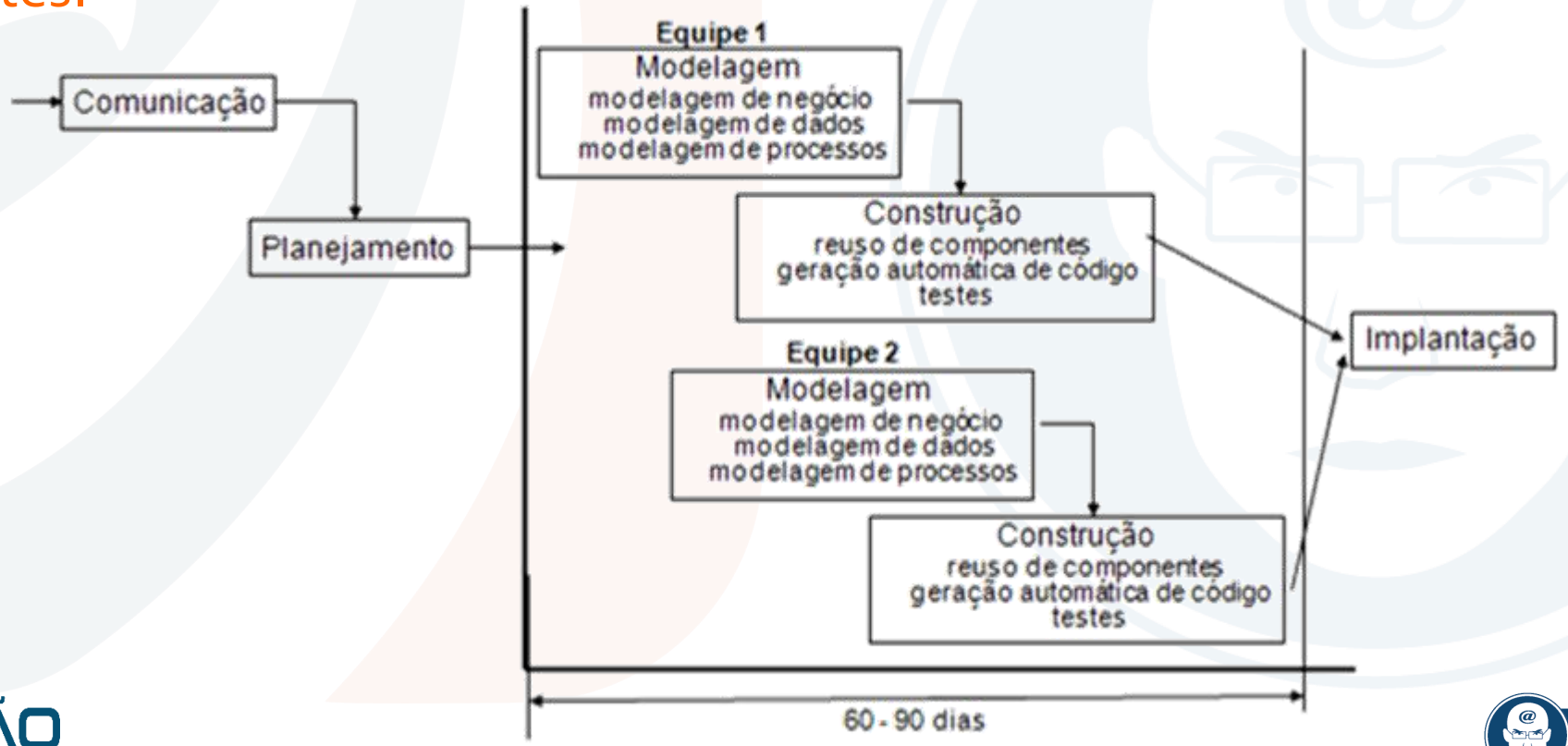


Modelos Incrementais

- O modelo incremental aplica sequencias lineares, de forma iterativa, à medida que o tempo avança. Cada sequencia linear (cascata) gera "incrementos" (entregáveis) do software. Seu foco consiste na entrega de um produto operacional a cada incremento.
- Vantagens:
  - Custo de acomodar mudanças de requisito é reduzido.
  - Facilita o feedback do cliente.
  - Todo incremento gera uma entrega.
- Desvantagens:
  - Processo não é visível.
  - A estrutura do sistema tende a degradar com a adição dos novos incrementos (exige refatoração).

Modelo RAD

- Rapid Application Development (RAD) é um modelo de processo de software incremental que enfatiza um ciclo de desenvolvimento curto.
- É uma adaptação "de alta velocidade" do modelo em cascata, no qual o desenvolvimento rápido é conseguido com o uso de uma abordagem de construção baseada em componentes.



Modelo RAD

- Desvantagens:
  - Projetos grandes precisam de muitas equipes.
  - Exige comprometimento dos clientes e desenvolvedores.
  - Sistemas não modularizados são problemáticos.
  - Não é adequado para projetos com grandes riscos técnicos.

[CESPE – BACEN – 2013] Na abordagem incremental, o sistema e dividido em subsistemas por funcionalidades. Após a identificação das funcionalidades a serem entregues no primeiro incremento, novas funcionalidades serão adicionadas a cada versão.

[CESPE – TRT/17 – 2013] Uma das desvantagens atribuídas ao modelo do desenvolvimento rápido de aplicação consiste na exigência da dedicação total do cliente e de desenvolvedores para a execução de tarefas constantes em um curto período de tempo.

Modelos Evolucionários

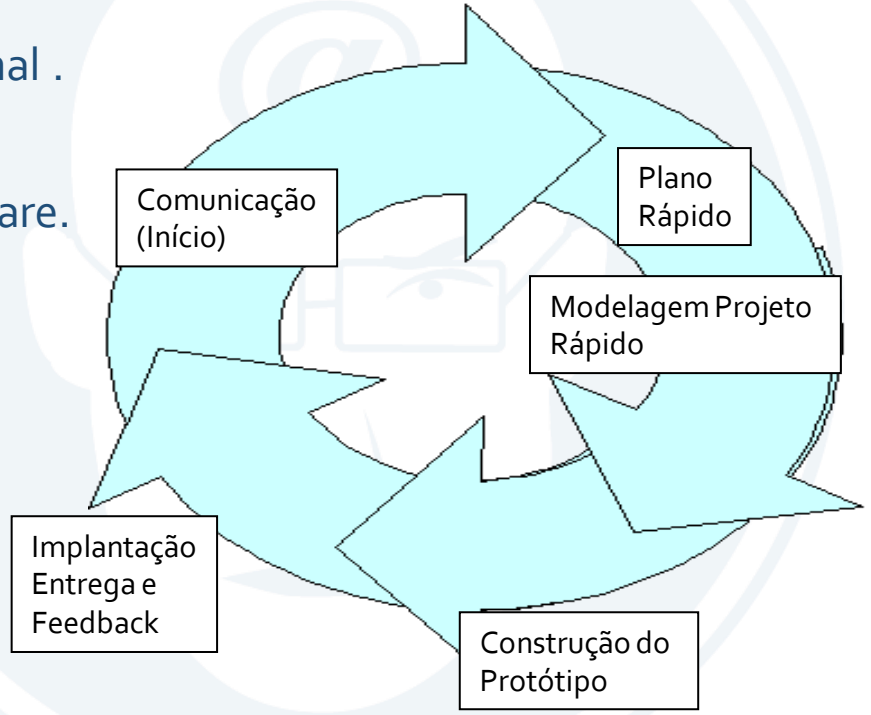
Modelos Evolucionários

- São modelos iterativos com características que possibilitam desenvolver versões cada vez mais complexas do software.
- São modelos evolucionários:
  - Prototipação.
  - Espiral.
  - Desenvolvimento Concorrente.

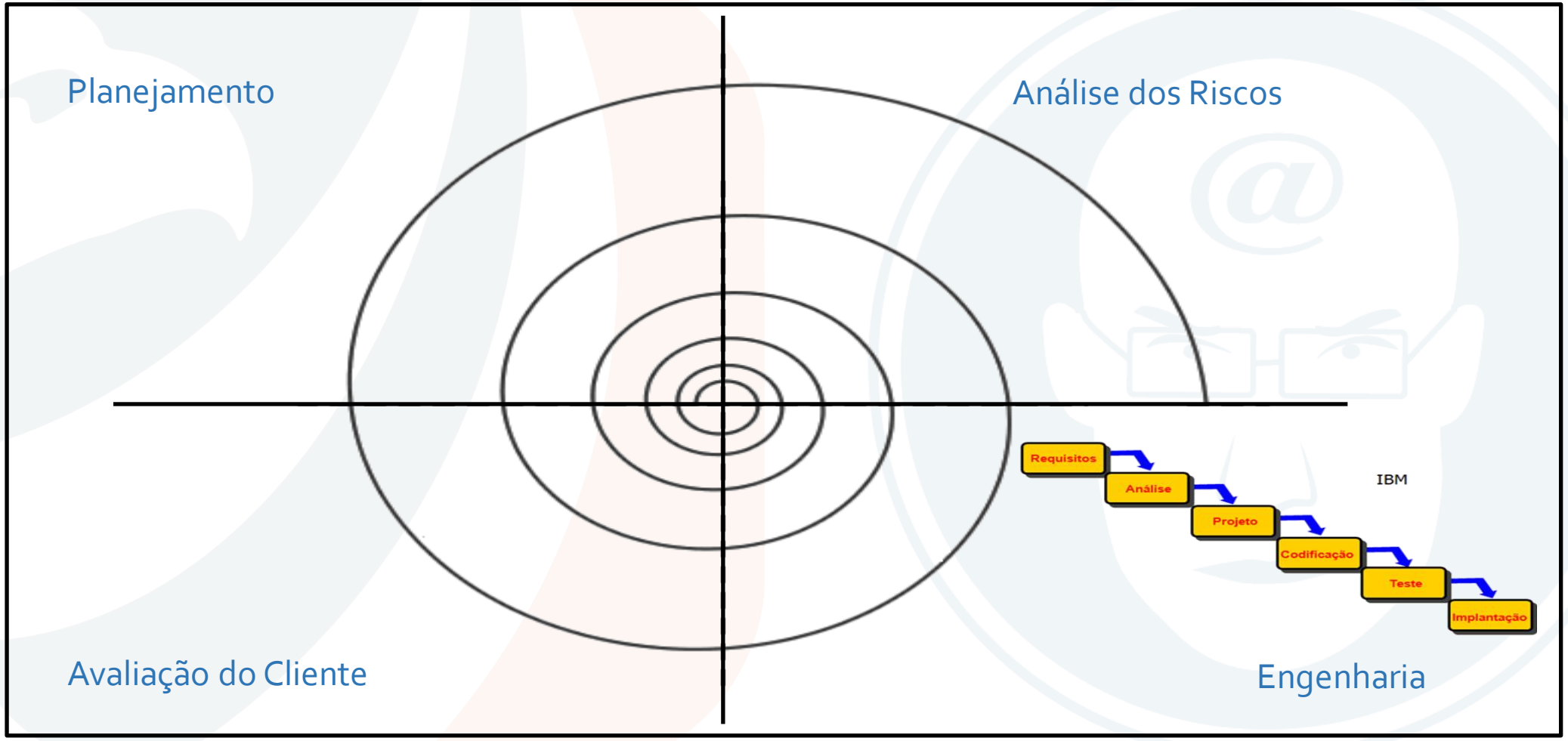
Obs: Os modelos incrementais e evolucionários são iterativos. A diferença entre os modelos incrementais e evolucionários é que os incrementais tem por objetivo apresentar um produto operacional a cada incremento.

Prototipação

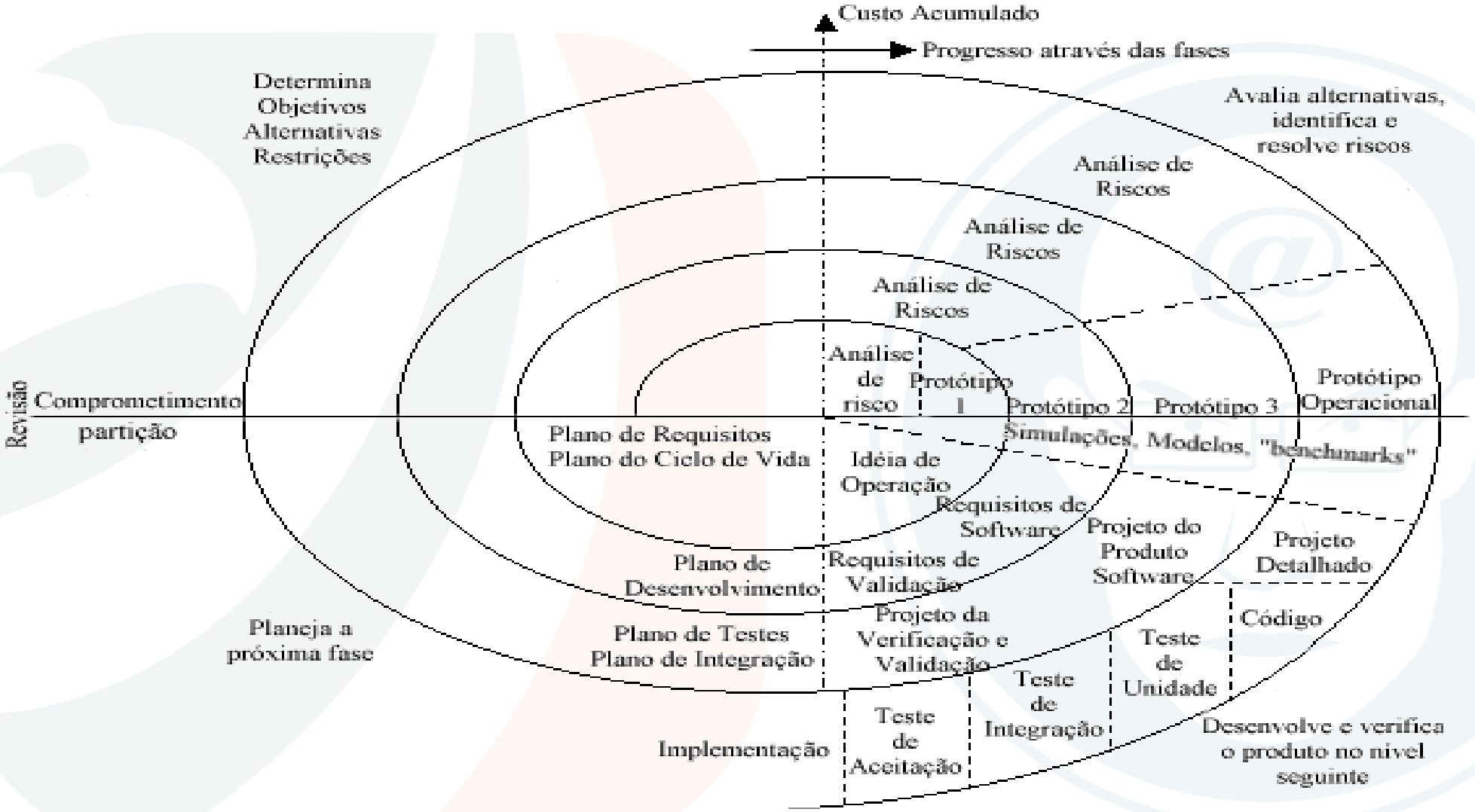
- Auxilia o engenheiro de software e o cliente a entenderem melhor o que deve ser construído quando os requisitos estão confusos.
- O protótipo serve como um mecanismo para a identificação dos requisitos de software.
- Desvantagens:
  - Os interessados enxergam o protótipo como um software operacional .
  - O engenheiro de software se compromete a colocar o software em produção rapidamente comprometendo a qualidade final do software.
- Tipos de protótipos:
  - **Protótipo exploratório,** que é descartado quando fica pronto, também chamado de protótipo para descarte.
  - **Protótipo evolutivo,** que gradualmente evolui para se tornar um sistema real.



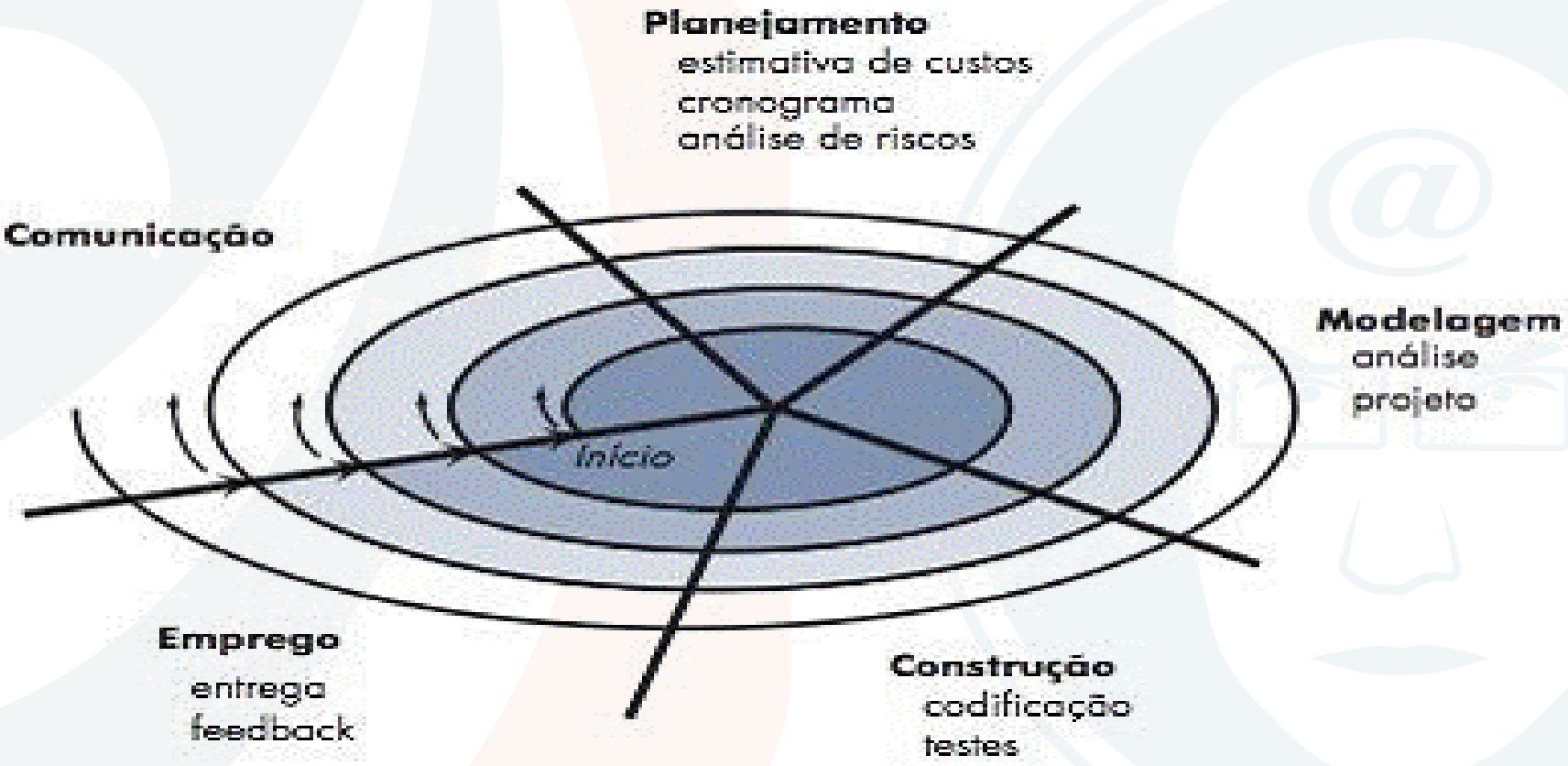
Espiral


Planejamento
Análise dos Riscos
Requisitos
Análise
Projeto
Codificação
Teste
Implantação
IBM
Avaliação do Cliente
Engenharia

Espiral (Boehm)



Espiral (Pressman)

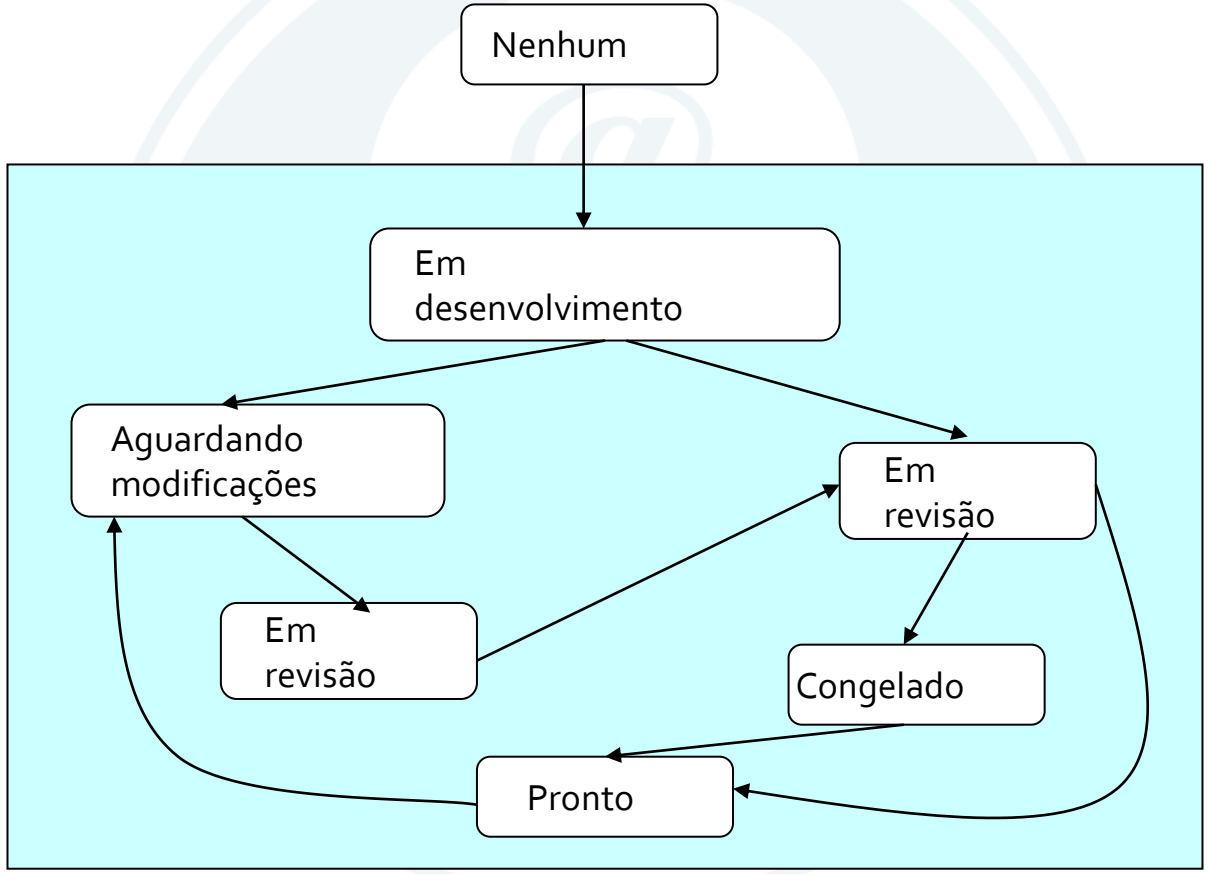


Espiral

- Este modelo foi proposto por Boehm em 1988, combina a natureza iterativa da prototipagem com os aspectos controlados e sistemáticos do modelo em cascata.
- O software é desenvolvido numa série de versões evolucionárias. As primeiras iterações podem gerar modelos em papel ou protótipos.
- Exige a consideração direta dos riscos técnicos em todos os estágios do projeto.
- Pode ser adaptado para aplicação ao longo da vida do software.

Desenvolvimento Concorrente

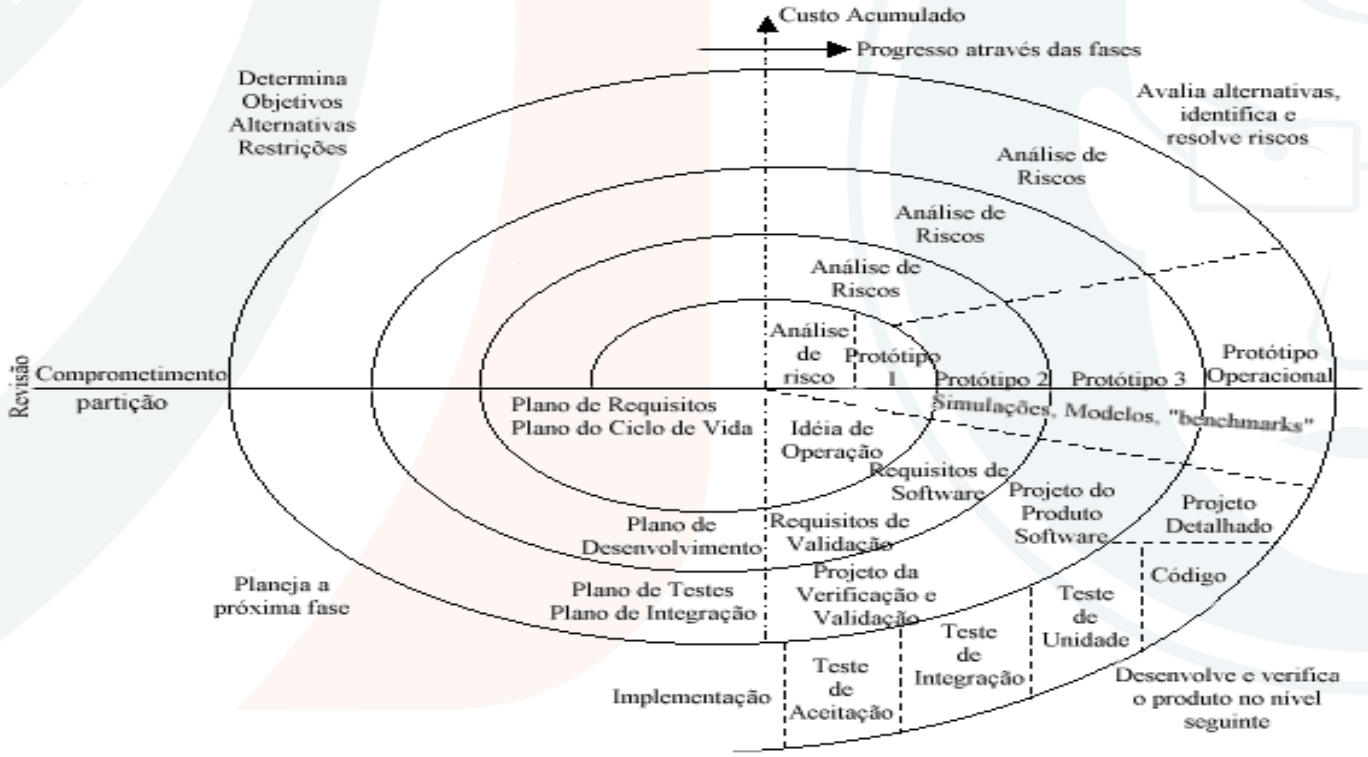
- É representado esquematicamente como uma série de atividades (levantamento de requisitos, análise, projeto, etc) e seus estados associados.
- Define uma série de eventos que vão disparar transições de estado para estado.
- É aplicável a todos os tipos de desenvolvimento de software e fornece uma imagem precisa do estado atual de um projeto.



[CESPE – IPHAN – 2018]

Situação hipotética: Uma organização implantou o framework de processo de software dirigido a riscos proposto por Boehm (modelo espiral), seguindo as atividades descritas pelo framework, em que a espiral é dividida em quatro setores. As atividades de criação do protótipo operacional, assim como os testes de integração e unitário, foram inseridas no setor de avaliação e redução de riscos.

Assertiva: Nessa situação, foi correta a inserção da atividade de protótipo no setor de avaliação e redução de riscos, porém as atividades de testes não poderiam ter sido inseridas, pois essas não são abrangidas por esse framework.



[CESPE – SUFRAMA – 2014] A prototipação pode ser utilizada para a verificabilidade de requisitos a partir da construção de um modelo do sistema a ser desenvolvido, bem como pode ser utilizada de forma combinada com a etnografia para compreender e obter requisitos.

[CESPE – TC/DF – 2014] Um protótipo de sistema auxilia na validação de requisitos, no projeto de interface com o usuário, podendo, ainda, ser usado para a realização de testes.

[CESPE – SERPRO – 2013] Mesmo não conseguindo capturar requisitos relacionados com o desempenho e a eficiência, os protótipos tem um bom emprego na elicitação e na validação dos requisitos de um sistema desejado pelo cliente.

[CESPE – SERPRO – 2013] A prototipação começa com a apresentação de um modelo de projeto rápido, etapa em que se representam as necessidades conhecidas e identificam-se as áreas que necessitam de maior definição, de modo a tornar claros os requisitos de um software.

[CESPE – TRF/10 – 2013] No ciclo de vida da primeira versão do modelo em espiral, a etapa de análise de riscos é realizada dentro da fase de desenvolvimento.

[CESPE – TRF/10 – 2013] No modelo prototipação, a construção de software tem várias atividades que são executadas de forma sistemática e sequencial.

[CESPE – TRT/17 – 2013] O modelo espiral de modelagem de processos para desenvolvimento de software é finalizado quando o software é implantado.

Modelos Especializados

Modelos Especializados

- Desenvolvimento Baseado em Componentes.
- Modelo de Métodos Formais:
  - baseados em modelos matemáticos;
  - lento e dispendioso;
  - necessita treinamento extensivo;
  - clientes despreparados não entendem os modelos, dificultando a comunicação.
- Desenvolvimento de Software Orientado a Aspectos:
  - gestão de memória;
  - integridade;
  - segurança.

Desenvolvimento Baseado em Componentes

- Oferece benefícios inerentes em qualidade de software, produtividade dos desenvolvedores e custos globais do sistema.
- É um processo que enfatiza o projeto e a construção de sistemas usando componentes de software reusáveis.
- Semelhante ao Modelo Espiral.
- Demanda um abordagem iterativa.
- Compõe aplicações a partir de componentes prontos.
- As atividades de modelagem e construção começam com a identificação de componentes candidatos.
- Os requisitos que não são atendidos pelos componentes são adaptados ou excluídos, quando possível.

[CESPE – ANATEL – 2014] Ao adquirir licença de uso de uma planilha eletrônica como ferramenta para automatização de escritório da empresa, está-se adotando uma engenharia de software orientada a reuso.

Engenharia Dirigida a Modelos (MDE)

Engenharia Dirigida a Modelos - MDE

- É uma abordagem do desenvolvimento de software segundo a qual os modelos, em vez de programas, são as saídas principais do processo de desenvolvimento. Os programas são gerados automaticamente a partir dos modelos.
- Aumenta o nível de abstração na engenharia de software, e, dessa forma, os engenheiros não precisam mais de preocupar com detalhes da linguagem de programação ou com as especificidades das plataformas de execução.
- Tem sua origem na Arquitetura Dirigida a Modelos (MDA), proposta pela OMG em 2001, como um novo paradigma de desenvolvimento de software.
- MDA se concentra nos estágios de Projeto e Implementação.
- MDE abrange todo o ciclo de desenvolvimento de software.

Engenharia Dirigida a Modelos - MDE

- Vantagens:
  - Permite que os engenheiros pensem em sistemas em alto nível de abstração, sem detalhes de implementação diminuindo a probabilidade de erros, acelerando o processo de projeto e implementação e permite a criação de modelos de aplicação reusáveis, independente de plataformas.
- Desvantagens:
  - As abstrações apoiadas pelo modelo nem sempre são corretas para a implementação.
  - Independência de plataforma são válidas apenas para sistemas de grande porte e duradouros, em que as plataformas se tornam obsoletas ao longo do tempo de vida do sistema.

Arquitetura Dirigida a Modelos - MDA

- Mecanismo de organização e gerenciamento de arquiteturas corporativas com suporte de ferramentas e serviços automatizados para a definição de modelos e realizar a transformação entre diversos tipos de modelos.
- Princípio básico: Separação clara entre lógica do negócio e tecnologias de implementação.

Arquitetura Dirigida a Modelos - MDA

- Três modelos:
  - Modelo de Computação Independente (CIM): modela as importantes abstrações do domínio usado no sistema. Podem ser chamados de "modelos de domínio".
  - Modelo Independente de Plataforma (PIM): modela a operação do sistema sem referência a sua implementação. O PIM geralmente é descrito por meio de modelos UML.
  - Modelo Específico de Plataforma (PSM): são as transformações do PIM com um PSM separado para cada plataforma de aplicação.

[CESPE – CNJ – 2013] 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.

[CESPE – CNJ – 2013] 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.

[CESPE – TRT/17 – 2013] Os estágios do ciclo de solução de um problema incluem situação atual, definição do problema, desenvolvimento técnico e integração da solução. Na engenharia de software, esses estágios não coexistem simultaneamente em nenhum nível de detalhe.

Gabarito

- [CESPE – SLU – 2019] C
- [CESPE – FUNPRESPE – 2016] E
- [CESPE – MPOG – 2015] E
- [CESPE – STJ – 2015] C
- [CESPE – STJ – 2015] E
- [CESPE – STJ – 2015] C
- [CESPE – ANTT – 2013] C
- [CESPE – ANTT – 2013] C
- [CESPE – TRT/10 – 2013] C
- [CESPE – TRT/10 – 2013] C
- [CESPE – TRT/10 – 2013] C
- [CESPE – STF – 2013] E
- [CESPE – STF – 2013] E
- [CESPE – SLU – 2019] E
- [CESPE – IPHAN – 2018] C
- [CESPE – IPHAN – 2018] E
- [CESPE – ANATEL – 2014] E

- [CESPE – BACEN – 2013] C
- [CESPE – TRT/17 – 2013] E

Gabarito

- [CESPE – IPHAN – 2018] E
- [CESPE – SUFRAMA – 2014] C
- [CESPE – TC/DF – 2014] C
- [CESPE – SERPRO – 2013] E
- [CESPE – SERPRO – 2013] E
- [CESPE – TRF/10 – 2013] E
- [CESPE – TRF/10 – 2013] E
- [CESPE – TRT/17 – 2013] E
- [CESPE – ANATEL – 2014] E

- [CESPE – CNJ – 2013] E
- [CESPE – CNJ – 2013] E
- [CESPE – TRT/17 – 2013] E

Fim