

# Aula 2: Modelos de Ciclo de Vida de Software

## Apresentação

---

Esta aula tem como tema central apresentar a você uma visão geral sobre medição e modelos de qualidade de software, usando boas práticas na implantação de programas de medição nas organizações.

A nossa motivação se baseia na preocupação em aumentar a qualidade, desempenho e produtividade, atributos fundamentais para toda organização desenvolvedora de software. Nesta abordagem, concluímos que as métricas estão diretamente ligadas ao controle dos processos de melhoria de software.

Pesquisas realizadas em empresas de softwares indicam que mais da metade de grandes projetos se depara com algum tipo de atraso, excesso de custo ou prazo ou fracasso na implantação. Isto se deve, principalmente, pela falta de controle nos projetos de software.

## Objetivos

---

- Definir o ciclo de vida de software;
- Listar modelos de ciclo de vida de software: cascata, evolutivo, iterativo e incremental;
- Discutir os modelos RAD, prototipação, espiral e RUP.

## Contextualização

---

O mundo moderno exige cada vez mais tecnologia e novas plataformas de gestão da informação e comunicação.

O software hoje está voltado não só para a web, mas também para diversos segmentos como indústria, comércio, saúde, segurança e muito mais. Além disso, atua na inteligência artificial, estando presente nos eletrodomésticos como TV, geladeiras, lavadoras, micro-ondas, sistema elétrico residencial e muitos outros.



(Fonte: TippaPatt / Shutterstock).

## Estudado dentro da área de Engenharia de Software, é considerado um fator fundamental para a obtenção de softwares de qualidade.

O processo de desenvolvimento de software está relacionado a pessoas, plataformas de gestão de qualidade e estruturas organizacionais, regidos por normas e procedimentos com o objetivo produzir softwares em conformidade com as exigências contratuais, limitados a aspectos como restrições, recursos, padrões etc.

### Por que é importante utilizar um modelo de ciclo de vida para desenvolver softwares?

**A resposta é simples:** o modelo de ciclo de vida é a primeira escolha a ser feita no processo de software, que abrange a vida do sistema, desde a definição de requisitos até a conclusão do projeto e entrega do produto.

#### Saiba mais

Esse ciclo de vida está agrupado em fases, como: definição de requisitos, análise, projeto, desenvolvimento, teste e implantação. Em cada fase são definidas, além das atividades, as funções e responsabilidades de cada membro da equipe do projeto.

É necessário conhecer as características do projeto a ser desenvolvido para escolhermos o melhor ciclo de vida e os recursos humanos empregados no projeto.

Esses devem ser reconhecidamente capazes de enfrentar os desafios que se apresentam diante das dificuldades inerentes de todo e qualquer projeto de software.

Principais modelos de ciclo de vida de software:

- Cascata;
- Modelo em V;
- Modelo incremental;
- Evolutivo;
- Iterativo e incremental;
- RAD;
- Prototipagem;
- Espiral;
- Modelo de ciclo de vida associado ao RUP.

Vamos ver esses modelos mais detalhadamente a seguir.

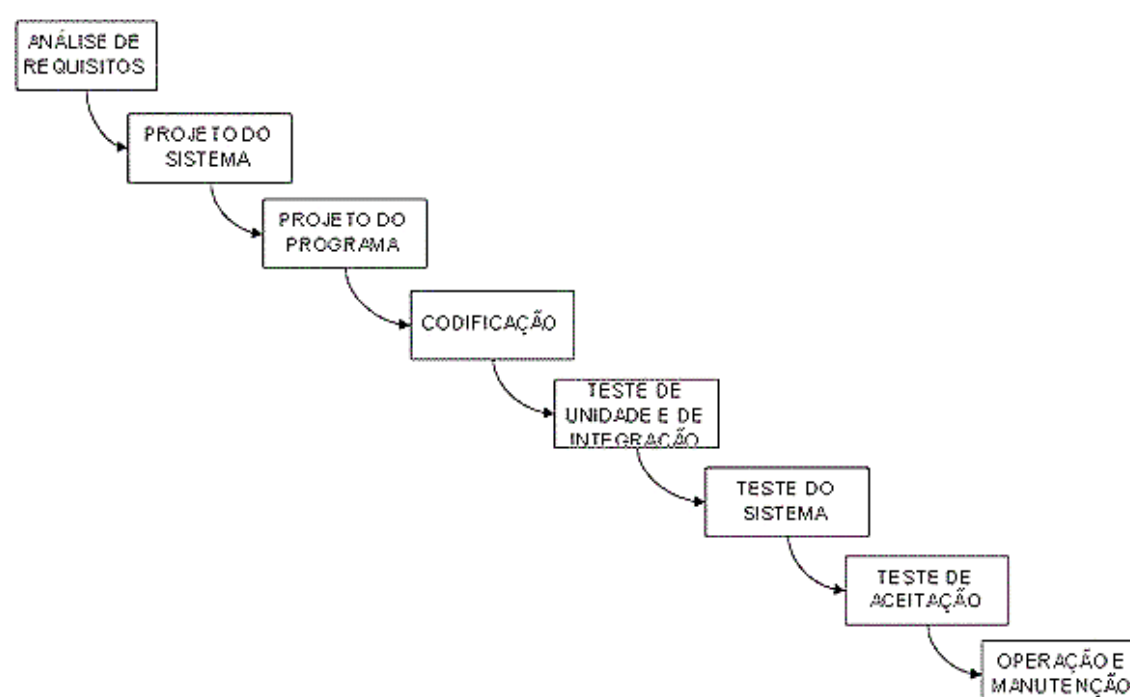
**Atenção!** Aqui existe uma videoaula, acesso pelo conteúdo online

## Cascata

O modelo cascata, ou sequencial linear, é considerado o mais antigo e o mais usado na engenharia de software em todo o mundo. Este é o grande mérito deste modelo.

Seu nome foi atribuído devido à sequência com que cada fase do desenvolvimento dependia do término da fase anterior, ou seja, o resultado de uma fase era utilizado com a entrada para a fase seguinte.

Nesse modelo, a ênfase maior ficava por conta das fases de análise e projeto antes de iniciar a implementação (codificação).



 **Figura 1:** Modelo cascata.

É possível retornar a uma etapa anterior; entretanto deve-se analisar muito bem em relação às áreas técnica e financeira, sem prejuízo das partes envolvidas.

### Vantagens do modelo cascata:

- A fase de análise de requisitos (única) leva ao projeto do sistema e, por sua vez, antes da fase de codificação;
- Ao final de cada fase e antes de iniciar a seguinte, devem ser feitas revisões com a participação do usuário;
- Cada etapa (fase) deve passar por aprovação e documentação, visando o passo seguinte.



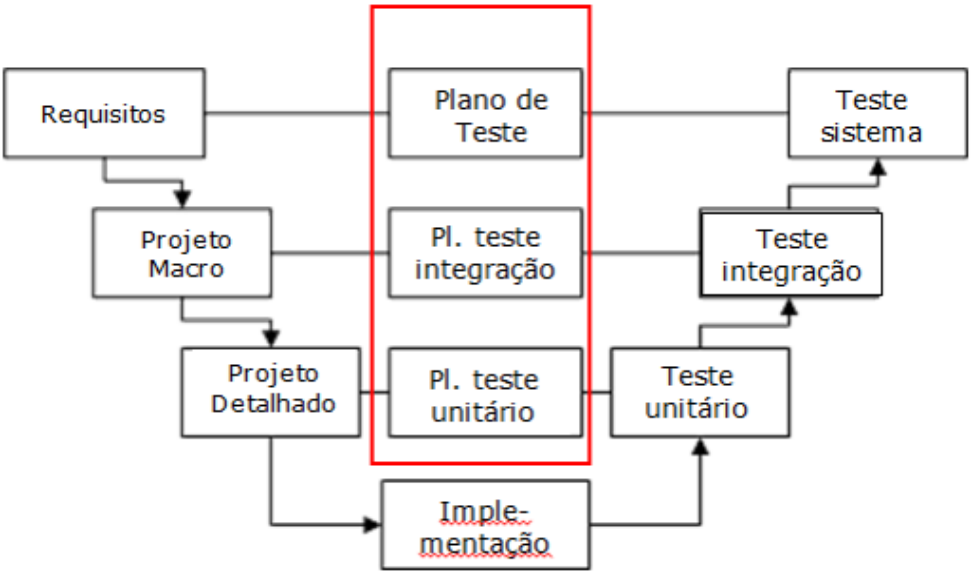
### Desvantagens do modelo cascata:

- O fluxo sequencial proposto pelo modelo geralmente não é seguido;
- O modelo cascata exige que os requisitos sejam claramente definidos e completos, pois são estabelecidos na fase inicial do projeto;
- O módulo executável para testes somente fica disponível ao final do projeto, isto é, na etapa avançada do desenvolvimento, exigindo um grande esforço.

## Modelo em V

Este modelo é uma variação do cascata, ajustando as atividades de testes com as fases de análise e projeto.

Da mesma forma que no cascata, o modelo em V é sequencial, o que significa que cada fase necessita ser concluída antes do início da fase seguinte.



**Figura 2:** Modelo em V.

Como podemos observar, no modelo em V os procedimentos de testes são planejados em praticamente todas as fases do desenvolvimento, diferentemente do que ocorre no modelo em cascata.

### Vantagens do modelo em V:

- Maior chance de sucesso quando comparado com o modelo cascata, devido ao desenvolvimento de planos de teste desde as fases iniciais;
- Funciona bem para projetos pequenos, nos quais os requisitos são facilmente entendidos.



### Desvantagens do modelo em V:

- Tão rígido quanto o modelo cascata;
- Requisitos devem ser estabelecidos corretamente e de forma clara no início do projeto.

# Incremental

No modelo incremental, as novas funcionalidades ou necessidades do usuário são integradas de forma segmentada e em série até a conclusão do produto final.

Este modelo se propõe a aumentar pouco a pouco o software, conforme as necessidades surgem.

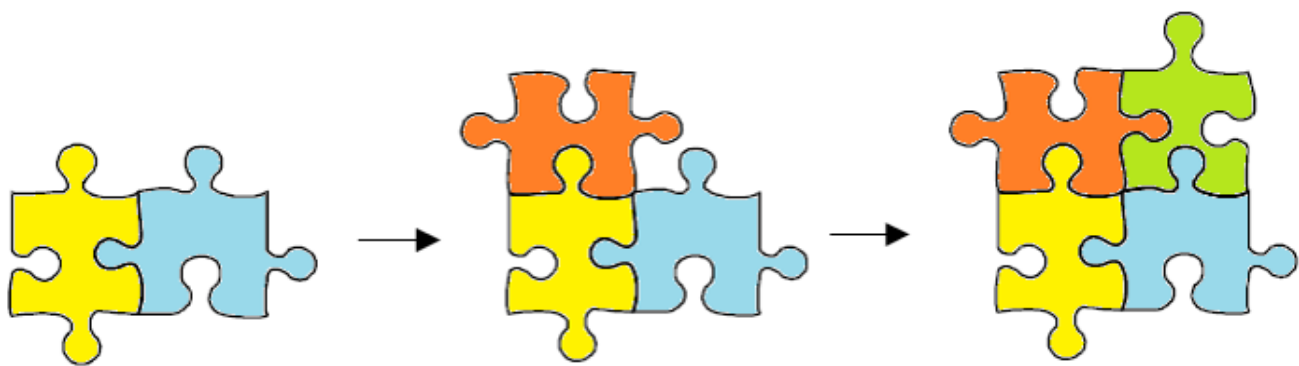


Figura 3: Modelo incremental.

O modelo incremental deve ser adotado quando os requisitos são claramente conhecidos na fase inicial do desenvolvimento e há a necessidade de entrega de um módulo ou funcionalidade do software em curto espaço de tempo.

## Atenção

É importante notar que, a cada incremento, uma nova versão do software é produzida.

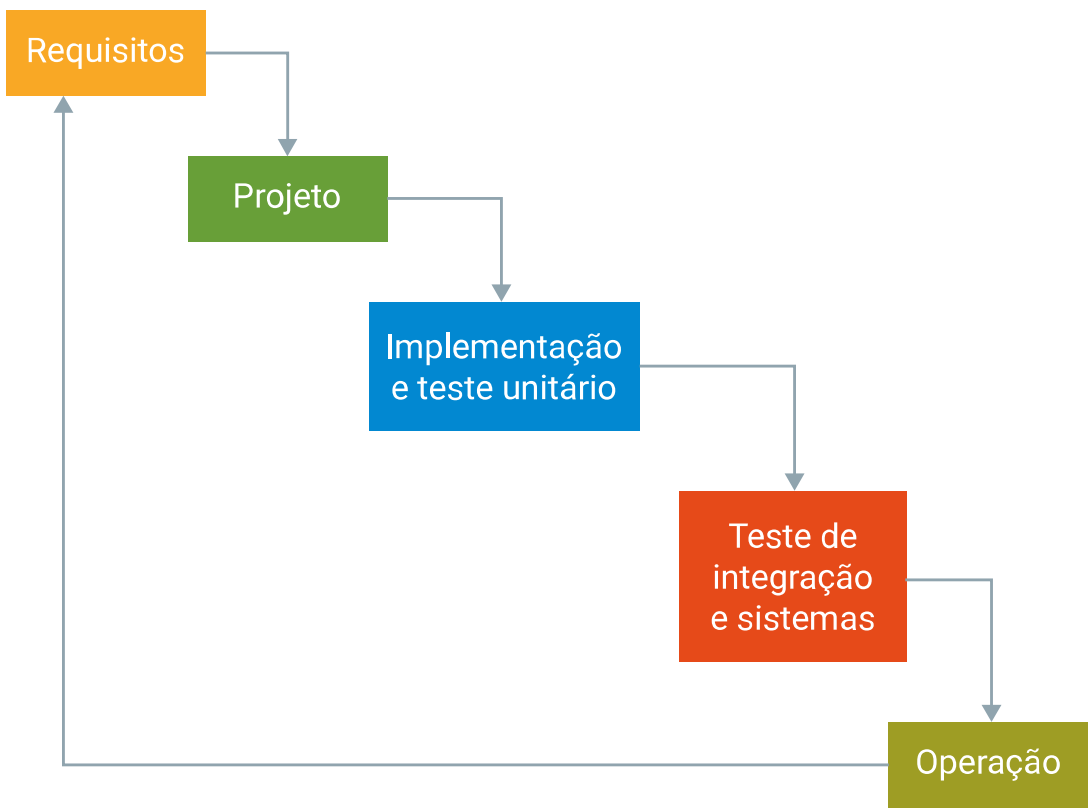


Figura 4: Fases do modelo incremental.

Vantagens do modelo incremental:

Desvantagens do modelo

- A entrega da primeira versão apresenta um menor custo e menos tempo se comparado com os modelos em cascata e em V;
- Poucos riscos associados ao desenvolvimento dos incrementos devido ao seu tamanho reduzido.



## incremental:

- No caso de requisitos mal definidos ou incompletos, alguns incrementos podem ser retrabalhados, o que gera custos e atrasos.

## Modelo evolutivo

---

O modelo evolutivo obedece a um processo que visa o desenvolvimento do software a partir de requisitos de protótipos iniciais.

Neste modelo, as versões parciais são desenvolvidas para atender aos requisitos levantados inicialmente. Assim, a primeira versão é usada para refinar os requisitos visando uma segunda versão e, a partir do conhecimento obtido com o uso, o desenvolvimento prossegue, evoluindo, então, o produto.

### Vantagens do modelo evolutivo:

- É usado quando os requisitos se apresentam incompletos no início do desenvolvimento;
- Com o uso do sistema, o conhecimento sobre o produto vai aumentando e, consequentemente, melhorando os requisitos.



### Desvantagens do modelo evolutivo:

- É necessário um rígido controle sobre os custos, grande preocupação com o cronograma e com a configuração do software;
- Usuários precisam estar preparados quanto aos resultados iniciais, que podem parecer insatisfatórios.

**Atenção!** Aqui existe uma videoaula, acesso pelo conteúdo online

## Iterativo e incremental

---

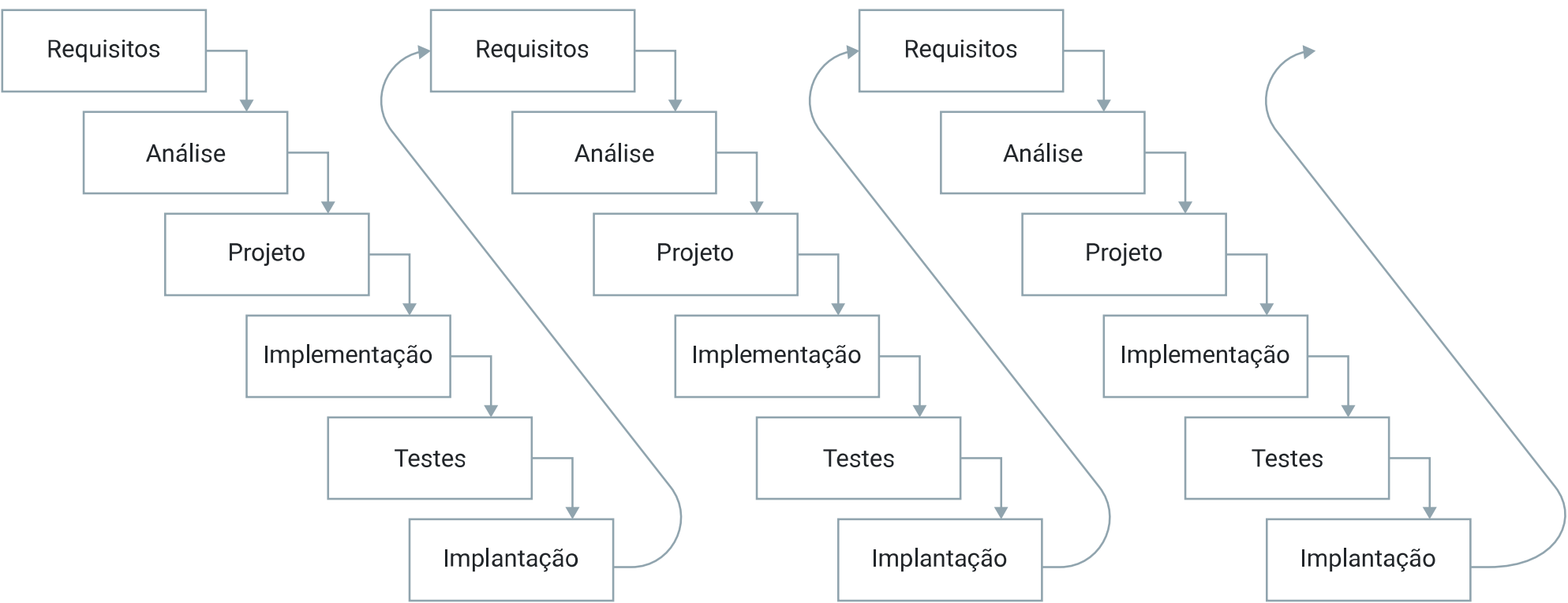
**O processo de desenvolvimento em cascata apresenta uma série de vantagens, porém tem também alguns pontos fracos. Para solucionar este impasse, estudiosos de TI criaram o modelo de desenvolvimento iterativo e incremental.**

Esse modelo trouxe bastante agilidade e flexibilidade nos processos de desenvolvimento de software, o que permitiu uma maior capacidade de lidar com as demandas do mercado e, assim, atingir melhores resultados em médio e longo prazo.

Basicamente, este modelo divide o desenvolvimento do software em pequenos ciclos ou módulos ou novas funcionalidades. Cada funcionalidade deve corresponder a uma necessidade do usuário à medida que for necessária à sua construção.

Para o desenvolvimento de cada funcionalidade, é utilizado um pequeno modelo em cascata, objetivando, assim, atender às necessidades sempre que se fizer necessário.

Precisamos, então, identificar as necessidades dos usuários e definir uma escala de prioridades para cada uma dessas funcionalidades.



 **Figura 5:** Modelo de desenvolvimento iterativo e incremental.

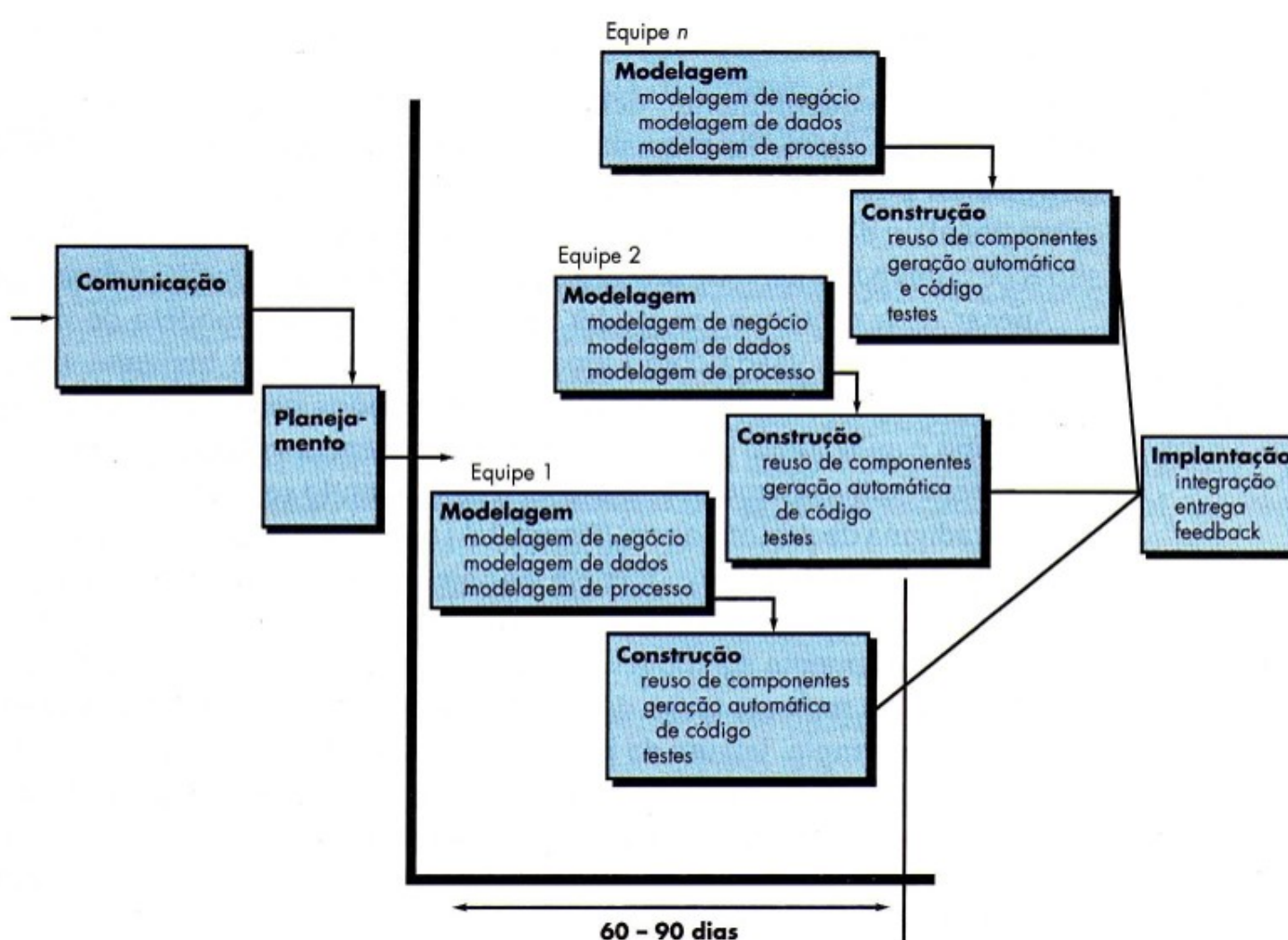
## RAD (*Rapid Application Development*)

**O modelo RAD é um modelo de desenvolvimento de software incremental que enfatiza um ciclo de desenvolvimento curto e se apresenta de forma sequencial linear.**

A rapidez ou velocidade com que o software é desenvolvido se deve pelo fato de várias equipes trabalharem em paralelo, quando o produto pode ser dividido em módulos. Um projeto de software que utiliza o RAD consome em média de 60 a 90 dias.

O uso deste modelo exige requisitos bem definidos e estabilizados, além de escopo restrito e com possibilidade de ser dividido em módulos.





**Figura 6:** modelo de desenvolvimento iterativo e incremental.

### Vantagens do modelo RAD:

- O ciclo de desenvolvimento é extremamente curto se comparado com os demais modelos;
- Distribuição de tarefas é facilitada entre as equipes de desenvolvimento.



### Desvantagens do modelo RAD:

- Requer equipes de desenvolvimento adequadas para atender à demanda de projetos grandes e escaláveis;
- Não é apropriado quando os riscos são grandes;
- Não é apropriado quando o sistema precisa interagir com outros sistemas.

## Prototipagem

O modelo prototipagem é utilizado quando é definido um conjunto de objetivos gerais para o software, mas os detalhes não estão claramente definidos como requisitos de entrada, processamento ou saída.

O ciclo do modelo prototipagem começa com a comunicação entre o engenheiro de software e o cliente, quando são definidos os objetivos gerais do software, identificando as necessidades conhecidas e são verificadas as áreas que necessitam de mais definição.

Os usuários podem fazer simulações com o protótipo, identificando mais detalhadamente funcionalidades que deverão ser disponibilizadas futuramente.



A versão inicial do software ajuda a determinar a viabilidade técnica, custos e prazo do projeto.

É importante haver uma forte interação com o usuário para que rapidamente o protótipo seja implementado de acordo com as etapas:



Figura 7: Etapas da prototipagem.

Vantagens do modelo prototipagem:

- O protótipo deve ser avaliado pelo usuário;
- A interação do usuário com o protótipo é fundamental para ajustar as necessidades funcionais.



Desvantagens do modelo prototipagem:

- O usuário pode pensar que a maior parte do software está pronta;
- O protótipo pode crescer de maneira não planejada, com risco de se tornar um incremento funcional;
- O protótipo pode confundir o usuário iludido por um desempenho melhor do que um incremento funcional;
- O fato de o protótipo não implementar toda a funcionalidade pode causar frustração para o usuário quando o sistema completo é entregue.

Modelo espiral

O modelo espiral se divide em duas etapas principais: análise de riscos e prototipagem. A cada novo ciclo, esse modelo testa constantemente erros que podem vir a acontecer.

A cada iteração, a volta da espiral pode ser baseada em um modelo diferente e pode ter diferentes atividades. Ou seja, a cada repetição é refeita a análise de a prototipação até que não existam grandes riscos no desenvolvimento.

No modelo espiral, em cada repetição do ciclo devem ocorrer no projeto:

Determinação de objetivos.

Avaliação e redução de riscos.

Desenvolvimento e validação.

Planejamento da próxima iteração.

Os riscos são considerados à medida que cada evolução é realizada.

A primeira atividade se dá com o desenvolvimento de uma especificação de produto, as próximas passagens podem ser usadas para desenvolver um protótipo e assim sucessivamente.

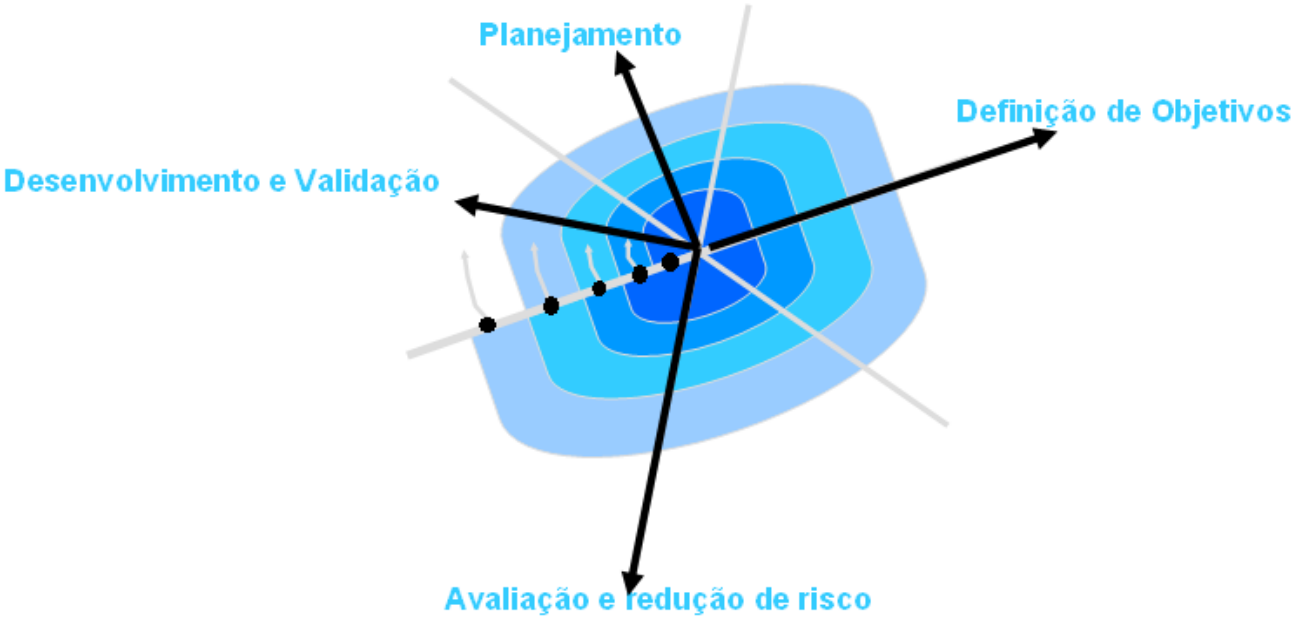


Figura 8: Modelo espiral.

Cada passagem pela fase do planejamento, por exemplo, resulta em ajustes no planejamento do projeto. O custo e o cronograma são sempre ajustados de acordo com o feedback obtido do usuário após uma entrega.

Na imagem, cada loop representa uma fase de desenvolvimento do software.

Vantagens do modelo espiral:

- Muita análise de riscos;
- Bom para projetos grandes e críticos;
- Software é produzido cedo no ciclo de vida.



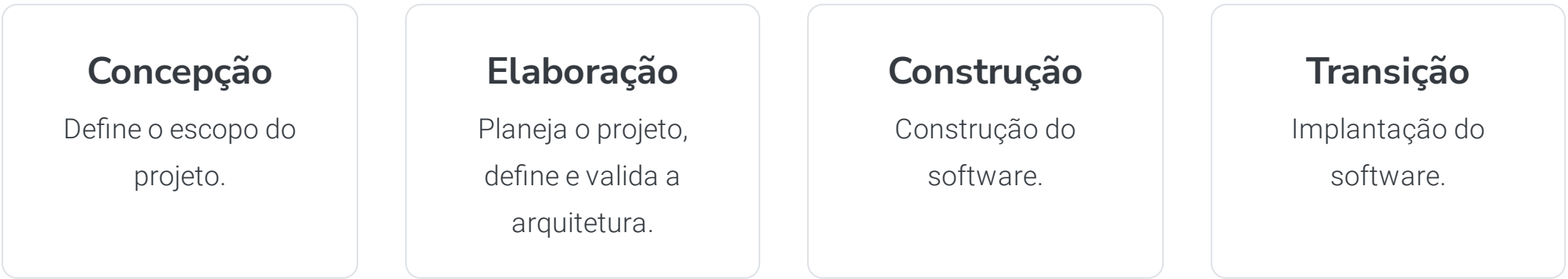
Desvantagens do modelo prototipagem:

- Pode ser custoso;
- Análise de riscos requer experiência;
- Não se aplica bem a projetos menores.

RUP (*Rational Unified Process*)

O RUP foi desenvolvido em 1999 por Jacobson, Booch e Rumbaugh depois de terem definido a UML. É um processo que integra ciclos, fases e disciplinas, visando a qualidade no gerenciamento de projetos.

O ciclo de vida do desenvolvimento do software é dividido nas seguintes fases:



As disciplinas requisitos, análise, projeto, implementação, testes e implantação vão estar presentes em cada uma dessas fases, com maior ou menor ênfase, conforma mostra a figura:

- REQUISITOS
- ANÁLISE
- PROJETO
- IMPLEMENTAÇÃO
- TESTES
- IMPLANTAÇÃO

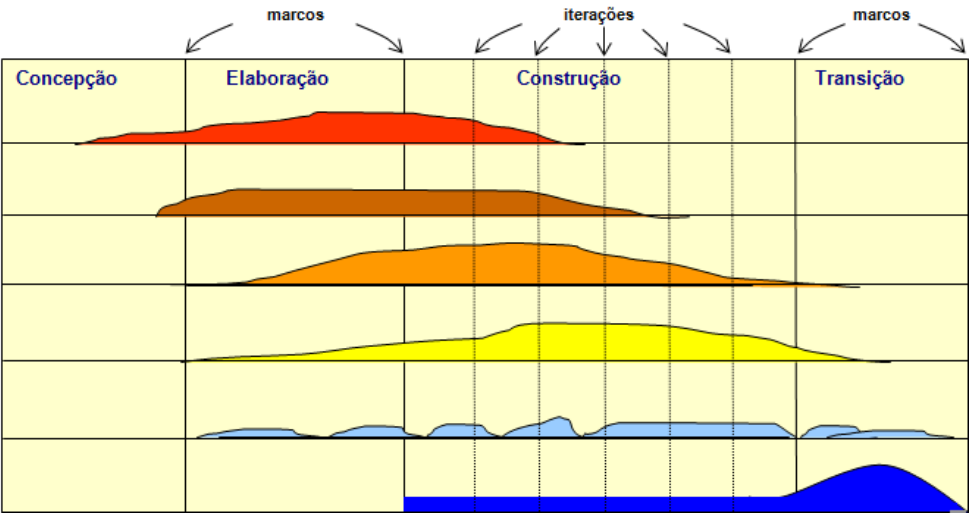


Figura 9: Disciplinas do modelo RUP.

# Atividade

---

1. (Fonte: FUNRIO 2013 – MPOG Analista de Tecnologia da Informação)

Considere o seguinte problema encontrado em projetos de desenvolvimento de software: projetos reais raramente seguem um fluxo sequencial. Apesar de um modelo linear poder acomodar a iteração, ele o faz indiretamente.

Como resultado, as modificações podem causar confusão à medida que a equipe de projeto prossegue. Esse é um dos problemas que são algumas vezes encontrados quando é aplicado qual modelo de desenvolvimento?

- a) Em cascata
  - b) Iterativo e incremental
  - c) Iterativo
  - d) Incremental
  - e) Evolutivo
- 

2. (Fonte: FCC 2013 – AL-RN Analista Legislativo – Analista de Sistemas)

O primeiro modelo de desenvolvimento de software a ser publicado foi derivado de processos mais gerais da engenharia de sistemas. Por causa do encadeamento entre uma fase e outra, esse modelo é conhecido como modelo em cascata ou ciclo de vida de software.

Dentre seus principais estágios, se encontra a análise e definição de requisitos, o projeto de sistema e software e...

- a) A análise de recursos e software
  - b) O desenvolvimento incremental
  - c) A geração de relatórios de teste
  - d) As pesquisas e testes
  - e) Implementação e teste unitário
- 

3. (Fonte: FUMARC 2012 – TJ-MG Oficial Judiciário – Assistente Técnico de Sistemas)

Em relação aos modelos de processos de software, pode-se dizer que os modelos incremental e evolucionário têm 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) Os incrementos de um software são entregues ao cliente somente duas vezes.
  - e) Um modelo de produção de software iterativo é composto pelas fases de análise de requisitos, projeto e implementação.
-

4. (Fonte: UFF 2009 – Analista de Tecnologia da Informação)

Em relação aos ciclos de vida do software, o desenvolvimento de sistemas por meio de ciclo de vida iterativos garante ao sistema:

- a) Atualização contínua
  - b) Legalidade
  - c) Segurança
  - d) Legibilidade
  - e) Utilização mínima de recursos
- 

5. (Fonte: CESPE 2010 – Detran-ES – Analista de Sistemas) Quando um aplicativo de software desenvolvido em uma organização atinge, no fim do seu ciclo de vida, a fase denominada aposentadoria, descontinuação ou fim de vida, todos os dados por ele manipulados podem ser descartados.

---

6. (Fonte: FCC 2013 – DPE-SP – Agente de Defensoria – Analista de Sistemas)

A prototipação representa uma técnica poderosa para o desenvolvimento de sistemas, mais especificamente do software desses sistemas. Sobre as funções desempenhadas por um protótipo, é correto afirmar que ele:

- a) Permite avaliar o desempenho geral da equipe de desenvolvimento de software.
  - b) Não permite que sejam realizados testes, visando verificar o funcionamento do sistema final, ainda que sejam testes parciais.
  - c) É inteiramente descartado, não sendo aproveitada nenhuma parte do código de software no sistema final entregue ao cliente.
  - d) Não possibilita avaliar a qualidade do software produzido.
  - e) Pode auxiliar na validação de requisitos do sistema, bem como propiciar a inserção de novos requisitos ainda não identificados.
- 

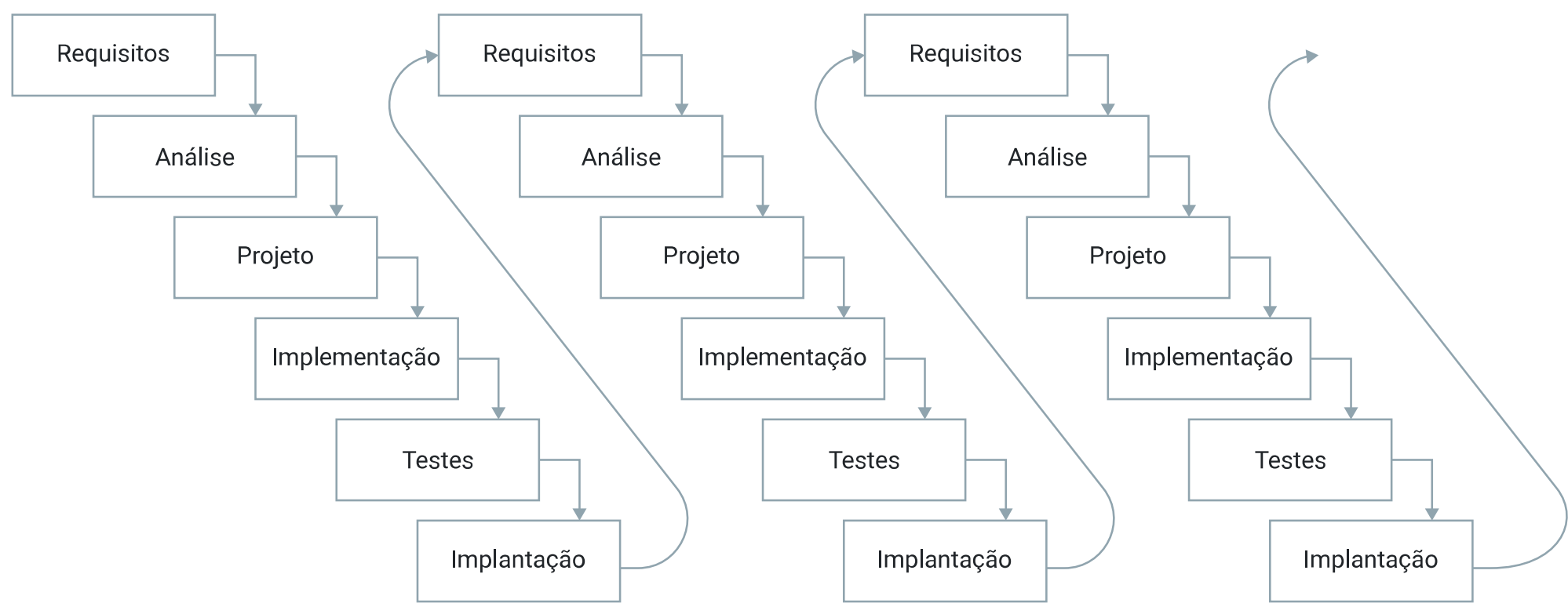
7. (Fonte: FCC 2012 – TST – Analista Judiciário – Analista de Sistemas)

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 softwares, existem alguns processos conhecidos.

Um desses processos tem 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 modelo de software em questão é:

- a) O modelo espiral
  - b) O modelo cascata
  - c) A prototipação
  - d) RAD
  - e) O modelo evolutivo
-

8. 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.



Esse 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
- e) Iterativo e evolutivo

9. (Fonte: IADES 2010 – CFA Analista de Sistemas)

O modelo espiral para a Engenharia de Software foi desenvolvido acrescentando-se novos elementos às melhores características de outros modelos. Segundo o modelo espiral, a determinação dos objetivos, alternativas e restrições está relacionada à atividade de:

- a) análise de risco
- b) planejamento
- c) engenharia
- d) avaliação feita pelo cliente
- e) feedback do cliente

10. No que se refere aos modelos de desenvolvimento e ciclos de vida, julgue o item que segue.

No modelo iterativo, divide-se o desenvolvimento em iterações. A cada iteração, podem ser acrescentadas novas funcionalidades ao software.

Uma iteração parte do estado no qual se encontravam os artefatos ao término da iteração anterior e resulta em um incremento. Uma iteração pode ter disciplinas como captura de requisitos, análise, projeto, implementação e teste.

---

11. Discuta como o modelo cascata e a prototipação podem ser utilizados no modelo em espiral.

## Referências Notas

---

PFLEEGER, Shari Lawrence. **Engenharia de Software**: teoria e prática. 2. ed. São Paulo: Pearson, 2003.

PRESSMAN, R. S. **Engenharia de Software**: uma abordagem profissional. 8. ed. Rio de Janeiro: Amgh, 2016.

SOMMERVILLE, Ian. **Engenharia de Software**. 9. ed. São Paulo: Pearson, 2011.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML**: guia do usuário. Rio de Janeiro: Campus, 2000.

GORDON, S.R.; GORDON, J.R. O ciclo de vida do desenvolvimento de sistemas. In: **Sistemas de informação**: uma abordagem gerencial. LTC, 2006. Disponível em: [http://www.pucrs.br/edipucrs/online/projetoSI/6-Engenharia/O\\_Ciclo\\_de\\_Vida\\_do\\_Development\\_de\\_Sistemas.pdf](http://www.pucrs.br/edipucrs/online/projetoSI/6-Engenharia/O_Ciclo_de_Vida_do_Development_de_Sistemas.pdf). Acesso em: 17 mai. 2019.

## Próxima aula

---

- Necessidade de buscar a qualidade nos testes de software;
- Características e utilização de testes adequados;
- Garantia para testes do software com o cliente.

## Explore mais

---

Pesquise na internet, sites, vídeos e artigos relacionados ao conteúdo visto. Em caso de dúvidas, converse com seu professor online por meio dos recursos disponíveis no ambiente de aprendizagem.

Leia os textos:

- [Ciclo de vida iterativo e incremental](#)
- [Conheça o Rational Unified Process \(RUP\).](#)
- [Comparação geral das metodologias clássicas de desenvolvimento de software](#)