



Engenharia de Software

ARA0097

Aula 02

Profª Otília C Nogueira

Aula 02

- O desenvolvimento de software como um projeto e as atividades de Gerenciamento;
- Planejamento e Gerenciamento de Riscos
- **Objetivos:**
 - Compreender as atividades de Gerenciamento de projetos que devem ser incorporadas ao processo de desenvolvimento de software.

Situação Problema

Se você deseja aprender um novo idioma e de vez em quando assiste algumas aulas na internet, ou faz algumas leituras esporádicas, isso pode ser considerado um projeto pessoal?

Suponha agora que você tenha planejado estudar duas horas por dia, três vezes na semana durante 18 meses, e após esse período irá fazer uma viagem de 30 dias para exercitar a conversação.

Isso pode ser considerado um projeto pessoal?

Mas antes vamos apresentar os modelos

- Modelo é uma descrição simplificada, uma abstração do processo de software
 - Um modelo de processo de software deve ser escolhido com base:
 - Na natureza do projeto e da aplicação;
 - Nos métodos e ferramentas a serem utilizados;
 - Nos controles e produtos que precisam ser entregues;

Processo de Software - Modelo

- Alguns modelos de processo:
 - Modelo Sequencial Linear;
 - Modelo RAD;
 - Modelos Evolucionários;
 - Modelo de Prototipação (Descartáveis)
 - Incremental (Exploratório)
 - Espiral (Exploratório)

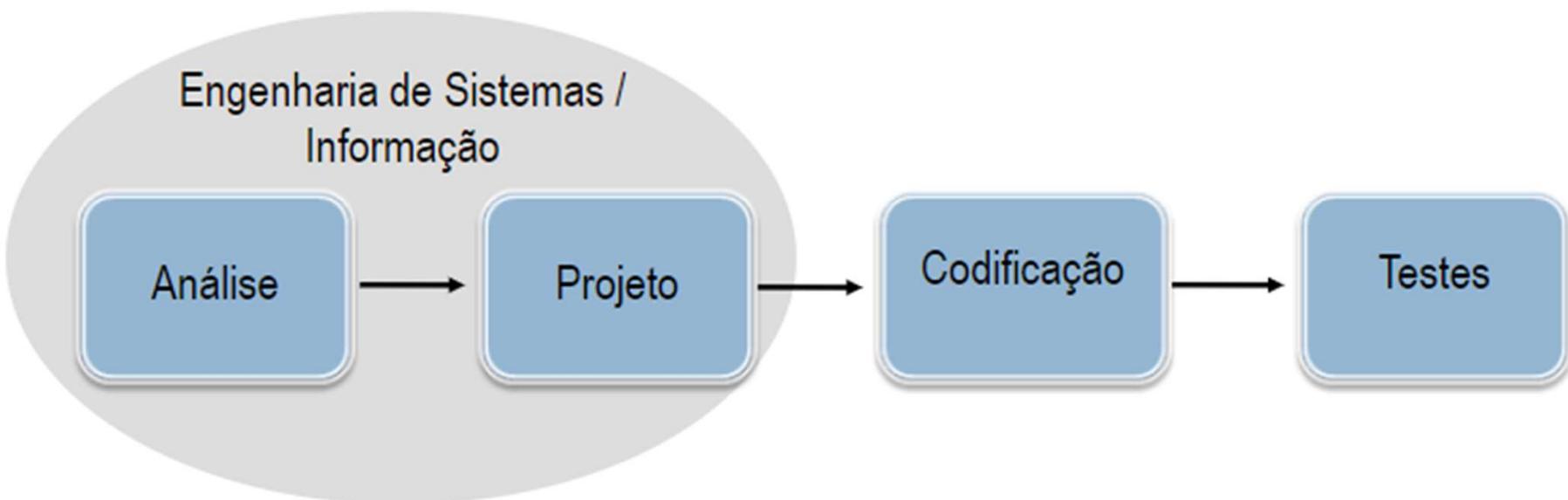
Modelo Sequencial Linear

- Também conhecido como ciclo de vida clássico ou Modelo Cascata:

- Modelo mais antigo e mais usado;
- Modelado em função do ciclo de engenharia convencional;
- Requer uma abordagem sistemática e sequencial para o desenvolvimento de um software

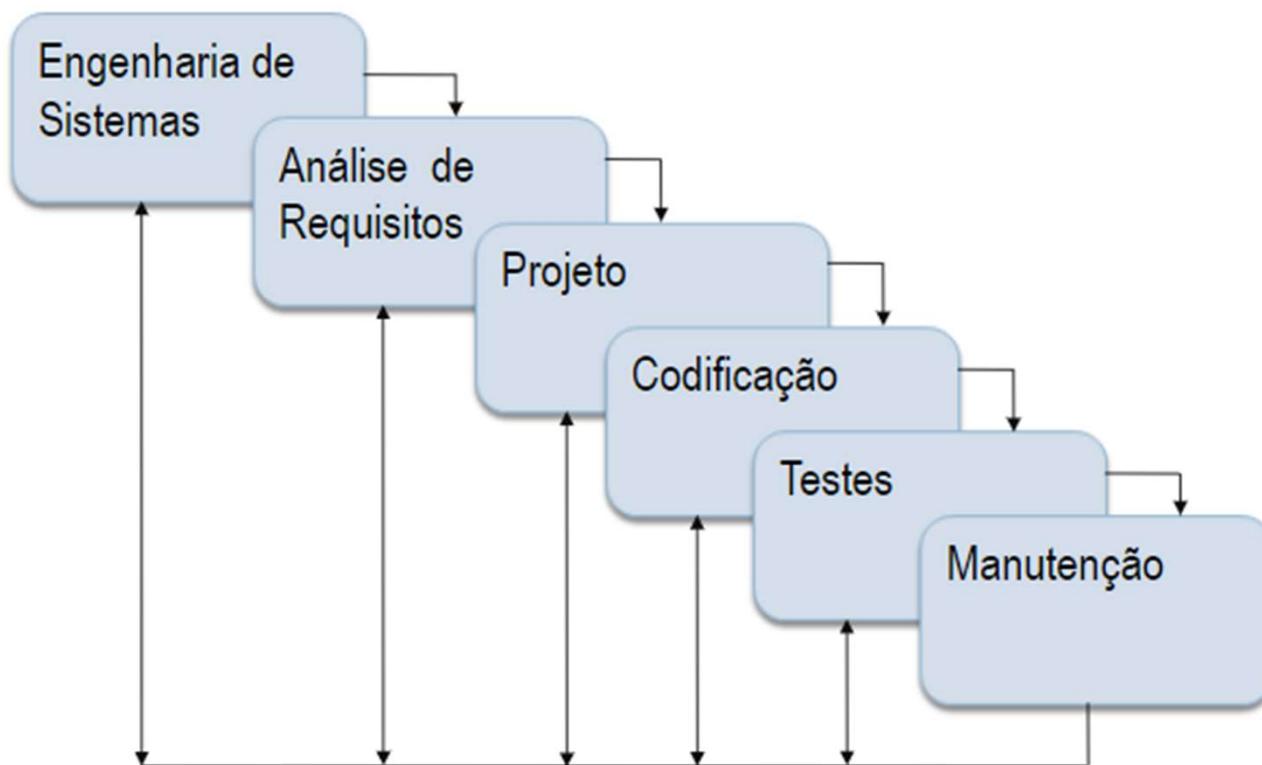
Modelo Sequencial Linear

- Muitos aplicam esse modelo de forma estritamente linear



Modelo Sequencial Linear

- Mas o modelo original prevê a volta:



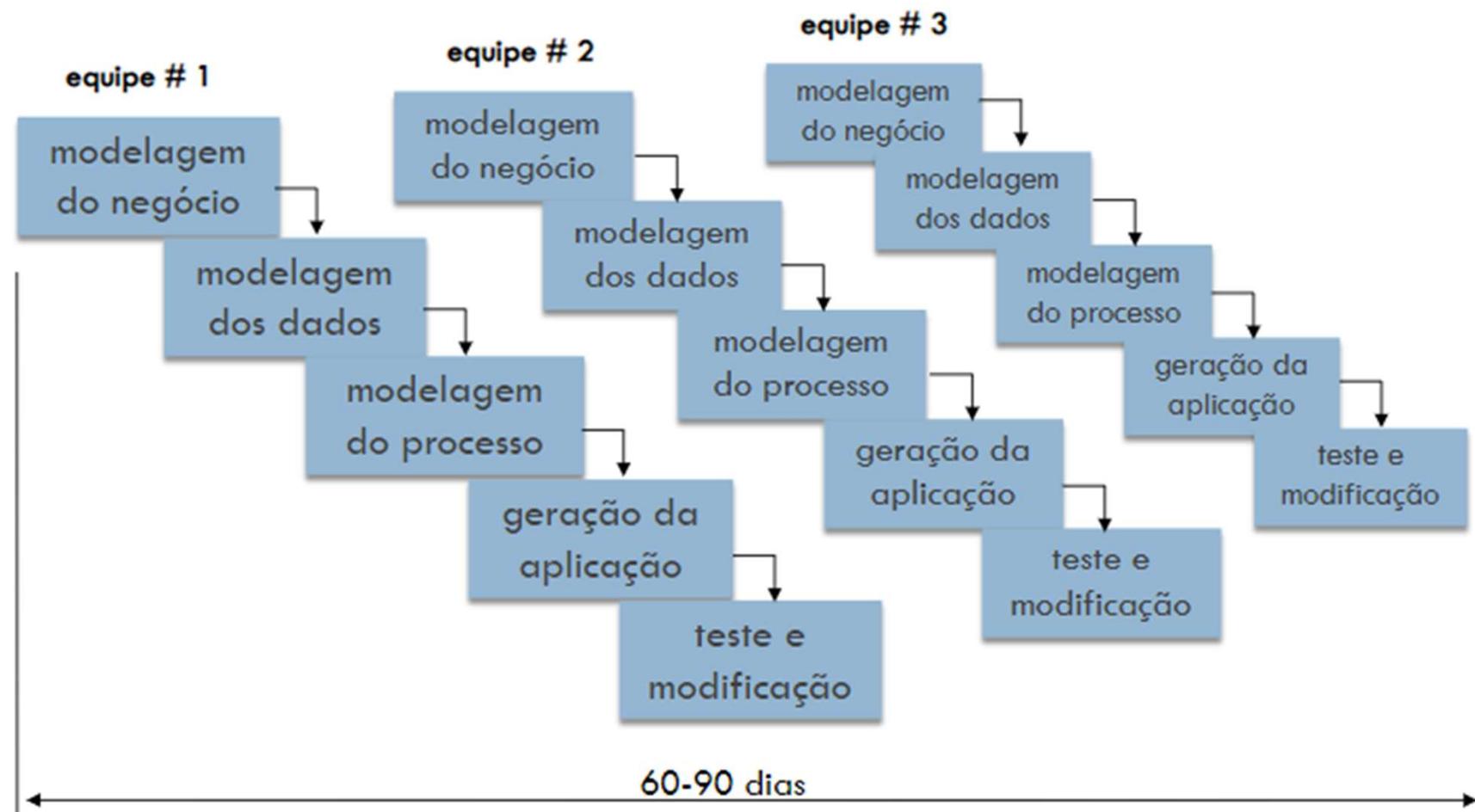
Modelo Sequencial Linear

- Problemas:
 - Projetos reais raramente seguem o fluxo sequencial proposto;
 - É difícil estabelecer todos os requisitos no começo do projeto na qual existe uma incerteza natural quanto a esses requisitos;
 - Uma versão do software só vai ficar pronto em um ponto tardio do desenvolvimento;
 - Assim se houver algum erro grave não detectado na análise ou projeto o resultado pode ser desastroso;
 - Há muitos estágios bloqueantes que permitem a ociosidade dos desenvolvedores em alguns momentos

Modelo RAD (Rapid Application Development)

- É o modelo seqüencial linear, mas que enfatiza um desenvolvimento extremamente rápido;
- A “alta velocidade” é conseguida através de uma abordagem de construção baseada em componentes;
- Usado quando os requisitos são bem definidos e o escopo do sistema é restrito.

Modelo RAD (Rapid Application Development)



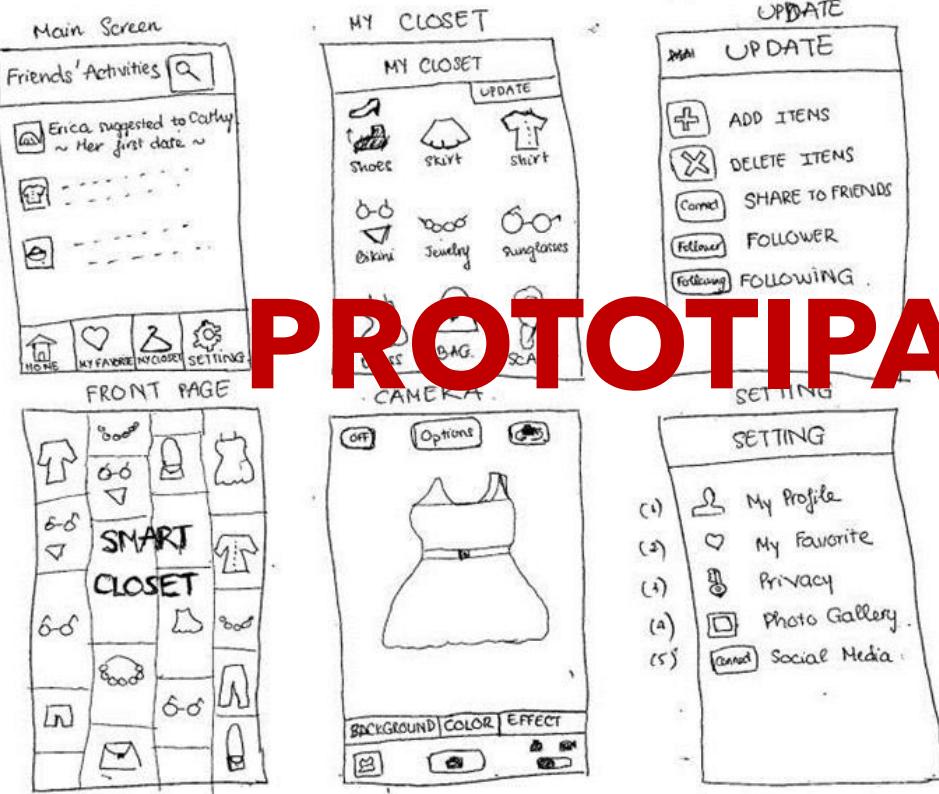
Modelo RAD

- Quando usar?
 - As restrições de tempo impostas pelo projeto demandam um escopo de escala.
 - Quando a aplicação pode ser modularizada de forma que cada grande função possa ser completada em menos de 3 meses.
 - Cada grande função pode ser alocada para uma equipe distinta e, depois são integradas para formar o todo.

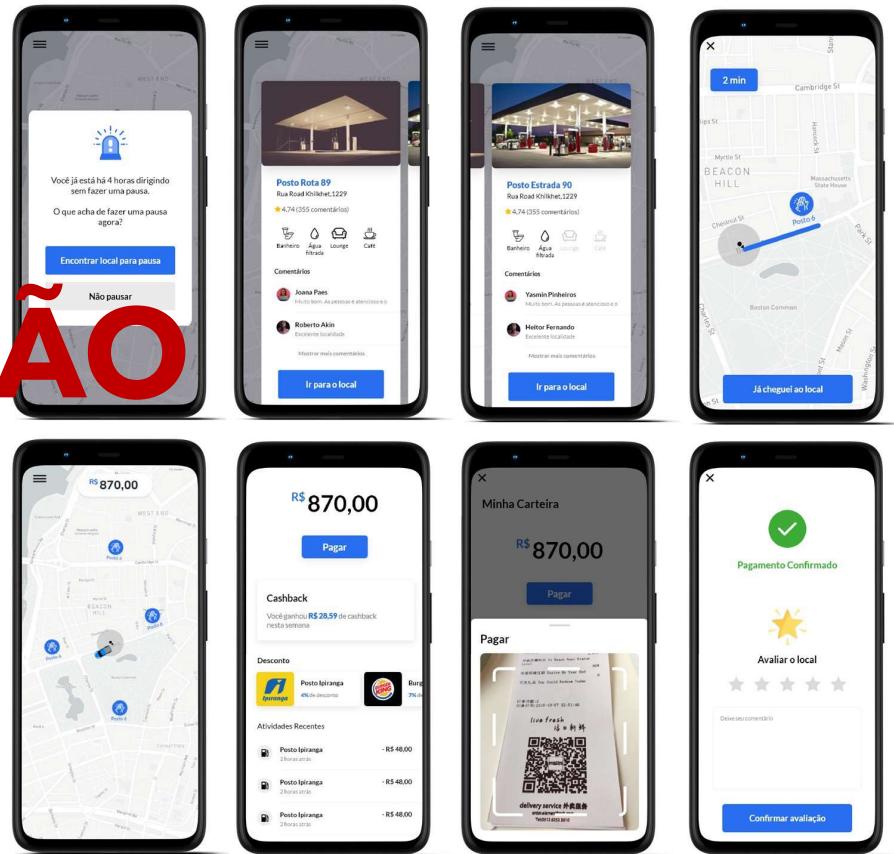
Modelo RAD

- **Problemas:**
 - Para projetos escaláveis, mas grandes, o RAD requer recursos humanos suficientes para criar um número adequado de equipes;
 - RAD requer um comprometimento entre desenvolvedores e clientes para que as atividades possam ser realizadas rapidamente e o sistema seja concluído em um tempo abreviado;
 - Se o comprometimento for abandonado por qualquer das partes, o projeto falhará;
 - Não é apropriado quando os riscos são grandes.

Modelos de Processos Evolucionários



PROTOTIPAÇÃO

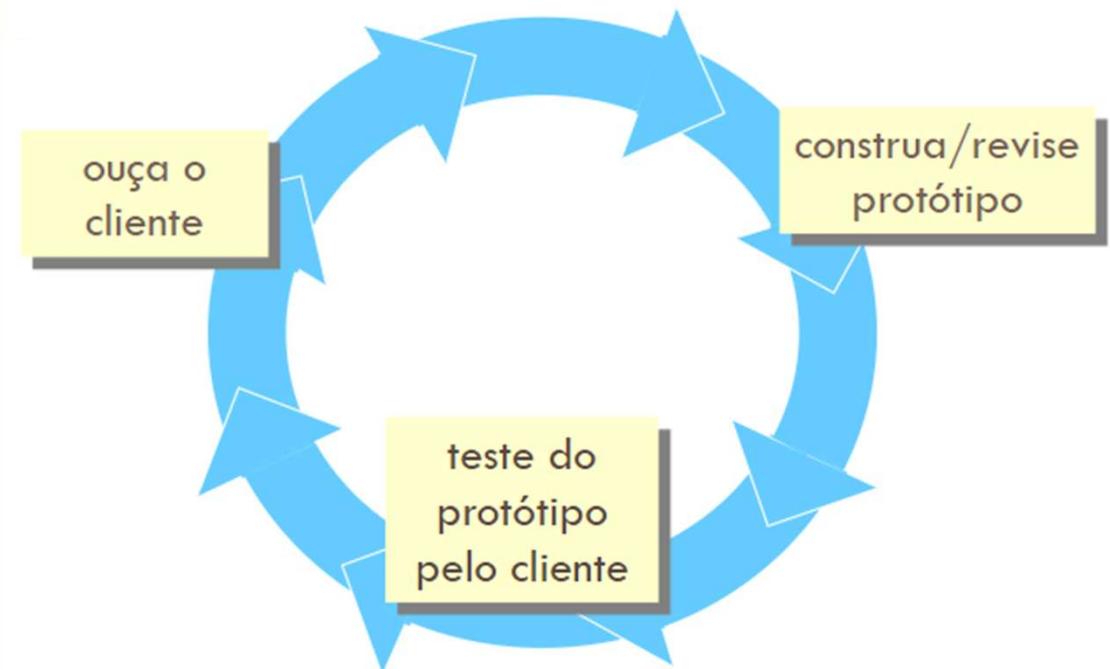
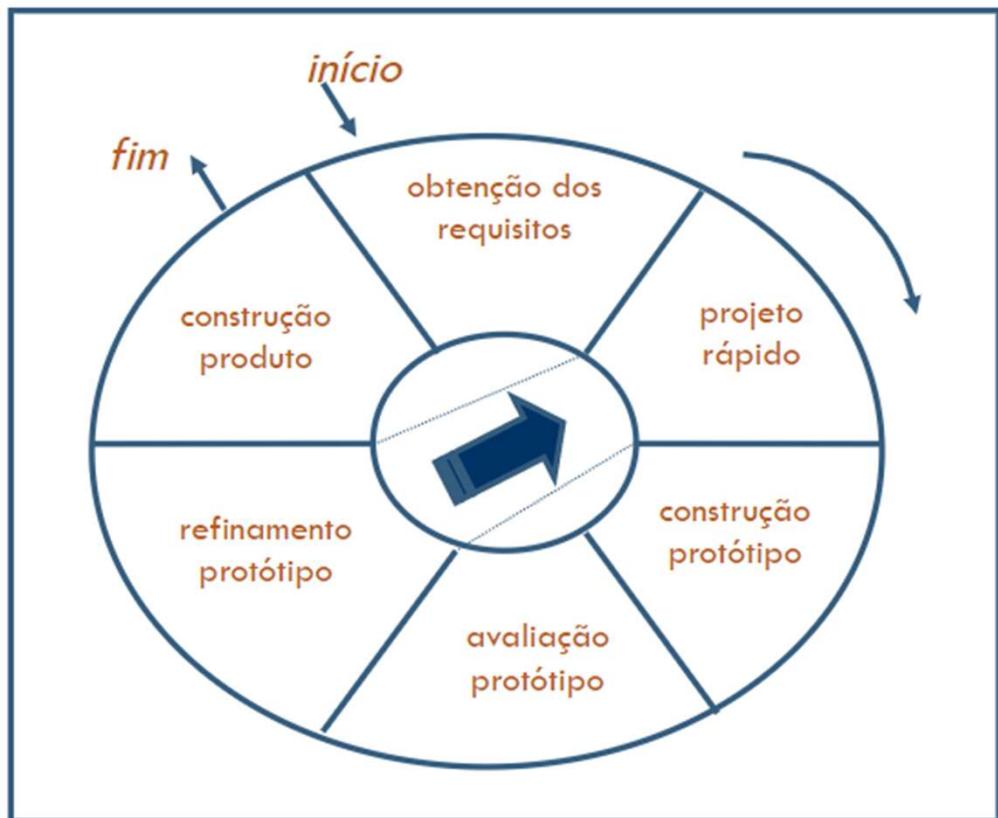


Prototipação

- Processo que possibilita que o desenvolvedor crie um modelo do software que deve ser construído.
- Idealmente, o modelo (protótipo) serve como um mecanismo para identificar os requisitos de software.
- Apropriado para quando o cliente definiu um conjunto de objetivos gerais para o software, mas não identificou requisitos de entrada, processamento e saída com detalhes.

Prototipação

Simplificando



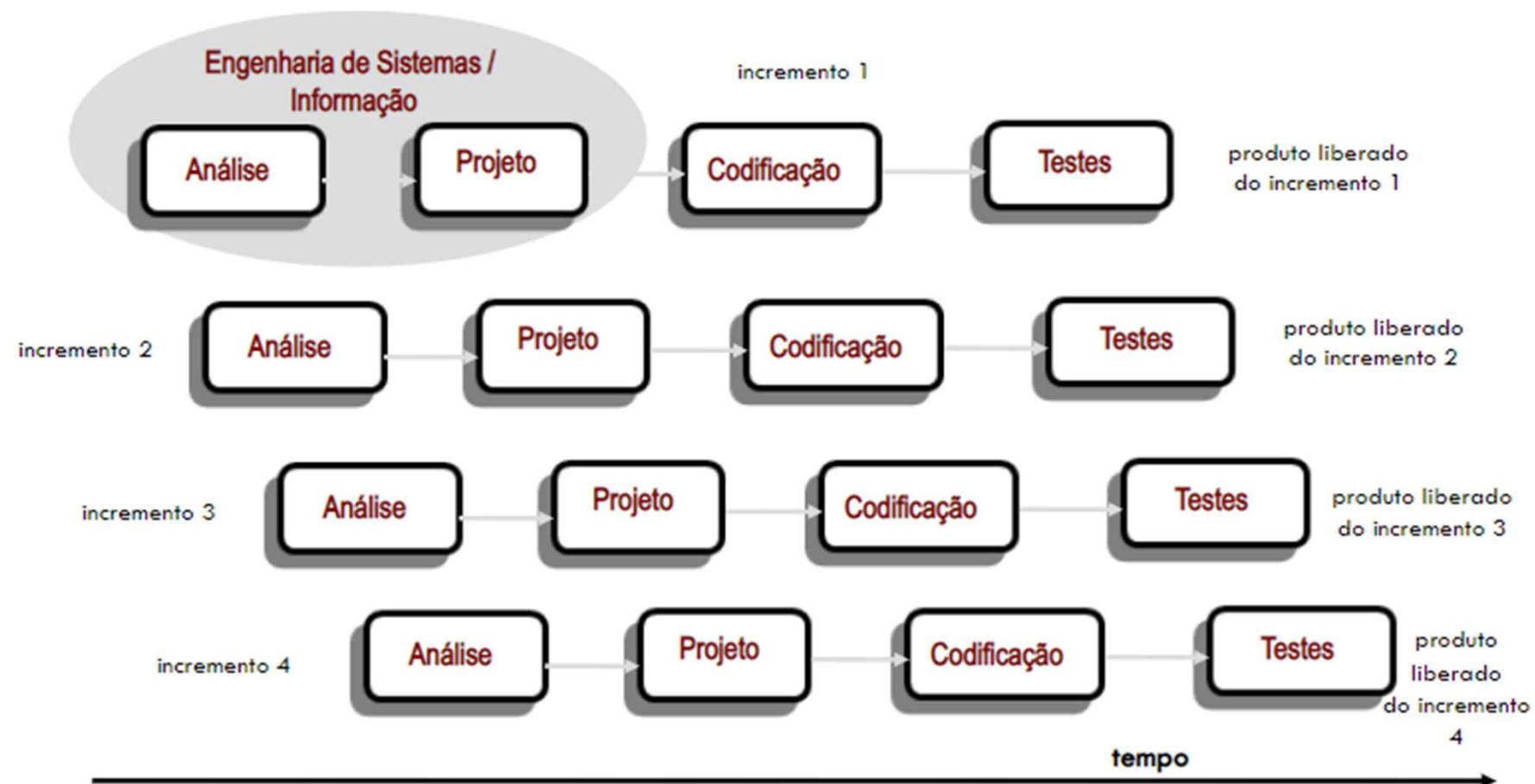
Prototipação

- Problemas:
 - Cliente não sabe que o software que ele vê, não considerou, durante o desenvolvimento, a qualidade global e a manutenibilidade a longo prazo. Não aceita bem a idéia de que a versão final do software vai ser construída e "força" a utilização do protótipo como produto final;
 - Desenvolvedor frequentemente faz uma implementação comprometida (utilizando o que está disponível) com o objetivo de produzir rapidamente um protótipo. Depois de um tempo ele se familiariza com essas escolhas, e esquece que elas não são apropriadas para o produto final.

Modelo Incremental

- Combina elementos do Modelo Linear com a filosofia da Prototipação.
- Aplica seqüências lineares numa abordagem de “saltos” à medida que o tempo progride
- Cada seqüência linear produz um incremento do software (proc. de texto)
- O processo se repete até que um produto completo seja produzido
- Difere da Prototipação pois a cada incremento produz uma versão operacional do software

Modelo Incremental



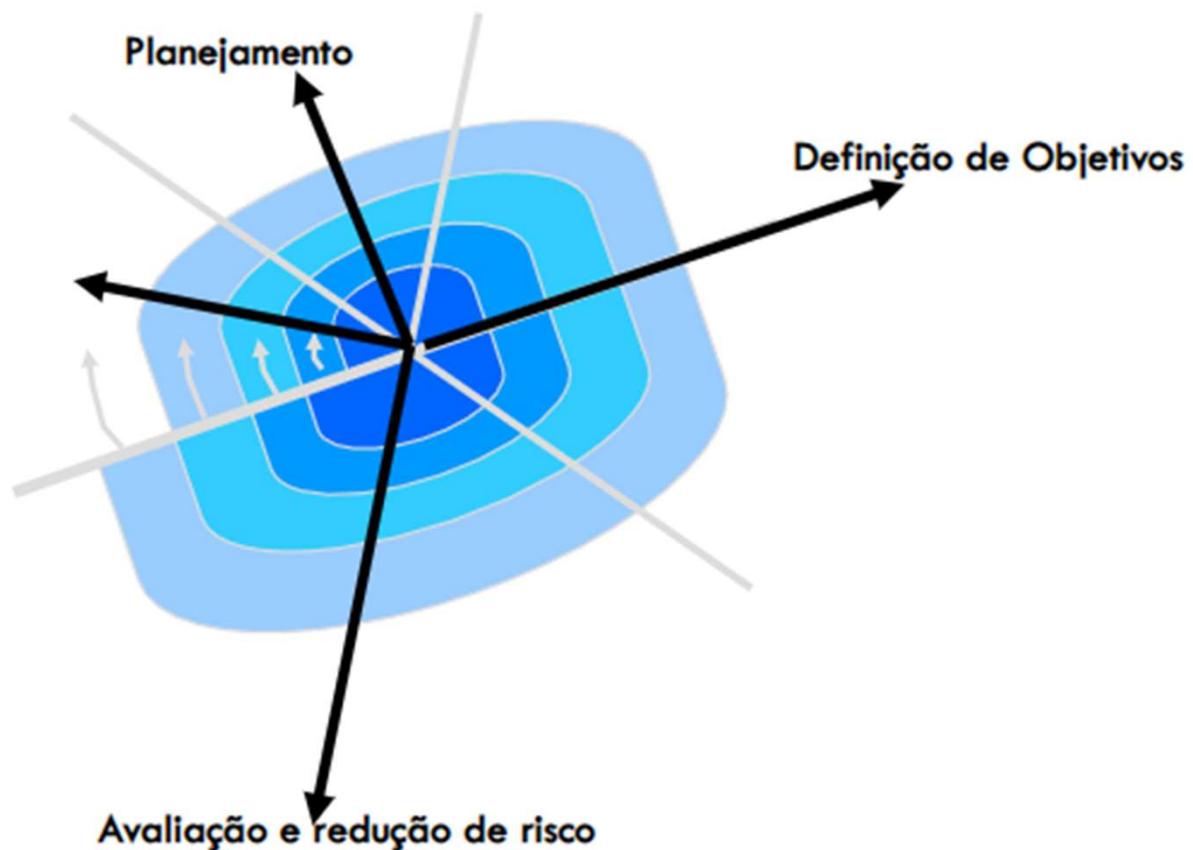
Modelo Espiral

- Engloba a natureza iterativa da **Prototipação** com os aspectos sistemáticos e controlados do **Modelo Linear**
- Fornece o potencial para o desenvolvimento rápido de versões incrementais do software
- Cada clico da espiral é uma fase do desenvolvimento que sempre passa por 4 aspectos

Modelo Espiral

Cada loop da espira é uma fase de desenvolvimento do software;

Desenvolvimento e Validação



Modelo Espiral - comentários

- É, atualmente, a abordagem mais realística para o desenvolvimento de software em grande escala.
- Usa uma abordagem que capacita o desenvolvedor e o cliente a entender e reagir aos riscos em cada etapa evolutiva.
- Pode ser difícil convencer os clientes que uma abordagem "evolutiva" é controlável.
- Exige considerável experiência na determinação de riscos e depende dessa experiência para ter sucesso.

Voltando ao nosso tema de hoje

Projeto

Leitura Específica

- Vídeo "PMBOK x ÁGIL qual a diferença" disponível em:
<https://www.youtube.com/watch?v=zFH4PNDZV0M>
- SOMMERVILLE, Ian. Engenharia de Software. 10^a Ed. São Paulo: Pearson Prentice Hall, 2011. Páginas 607 até 615.
Disponível em:
<https://plataforma.bvirtual.com.br/Leitor/Loader/168127/pdf>

O que é um projeto

- Para o **PMBOK**, projeto é um esforço temporário para criar um serviço ou produto ou resultado exclusivo. Para tal necessita de objetivos claros, parâmetros de medição, datas de início e término que atendam os requisitos das partes interessadas.

Projetos

projeto

temporário

Todo projeto
tem início e fim
definidos

progressivo

As características
do projeto são
detalhadas na
medida em que é
maior o
entendimento

único

O produto ou
serviço gerado
é diferente
sempre.

PMBOK

O Project Management Body of Knowledge, também conhecido como PMBOK® é um conjunto de práticas em gerência de projetos levantado pelo Project Management Institute (PMI) e constituem a base da metodologia de gerência de projetos do PMI. Estas práticas são compiladas na forma de um guia, chamado de Guia do Conjunto de Conhecimentos em Gerenciamento de Projetos, ou Guia PMBOK.



Exemplos de projetos

- lançamento de um novo produto ou serviço;
- construção de uma casa;
- desenvolvimento de um software;
- implantação de uma nova tecnologia;
- realização de uma viagem;
- publicação de um livro;
- organização de um evento (festa, reunião, congresso etc.);
- planejamento e implementação de uma mudança organizacional;
- construção de uma ponte.

GERENCIAMENTO DE PROJETO

4. GERENCIAMENTO DE INTEGRAÇÃO

- 4.1 Desenvolver o termo de abertura do projeto;
- 4.2 Desenvolver o plano de gerenciamento do projeto;
- 4.3 Orientar e gerenciar a execução do projeto;
- 4.4 Monitorar e controlar o trabalho do projeto;
- 4.5 Realizar o controle integrado de mudanças;
- 4.6 Encerrar o projeto ou fase;

8. GERENCIAMENTO DE QUALIDADE

- 8.1 Planejar o gerenciamento da qualidade;
- 8.2 Realizar a garantia da qualidade;
- 8.3 Controlar a qualidade;

12. GERENCIAMENTO DE AQUISIÇÕES

- 12.1 Planejar o gerenciamento das aquisições;
- 12.2 Conduzir aquisições;
- 12.3 Controlar as aquisições;
- 12.4 Encerrar as aquisições;

5. GERENCIAMENTO DO ESCOPO

- 5.1 Planejar o Gerenciamento do Escopo;
- 5.2 Coletar os Requisitos;
- 5.3 Definir o escopo;
- 5.4 Criar a EAP;
- 5.5 Validar o escopo;
- 5.6 Controlar o escopo;

6. GERENCIAMENTO DO TEMPO

- 6.1. Planejar o Gerenciamento do Cronograma;
- 6.2 Definir as atividades;
- 6.3 Sequenciar as atividades;
- 6.4 Estimar os recursos das atividades;
- 6.5 Estimar a durações das atividades;
- 6.6 Desenvolver o cronograma;
- 6.7 Controlar o cronograma;

9. GERENCIAMENTO DA RECURSOS HUMANOS

- 9.1 Planejar o gerenciamento dos recursos humanos;
- 9.2 Mobilizar a equipe do projeto;
- 9.3 Desenvolver a equipe do projeto;
- 9.4 Gerenciar a equipe do projeto;

10. GERENCIAMENTO DAS COMUNICAÇÕES

- 10.1 Planejar o gerenciamento das comunicações;
- 10.2 Gerenciar as comunicações;
- 10.3 Controlar as comunicações;

7. GERENCIAMENTO DE CUSTOS

- 7.1 Planejar o gerenciamento dos custos;
- 7.2 Estimar os custos;
- 7.3 Determinar o orçamento;
- 7.4 Controlar os custos;

11. GERENCIAMENTO DE RISCOS

- 11.1 Planejar o gerenciamento dos riscos;
- 11.2 Identificar os riscos;
- 11.3 Realizar a análise qualitativa dos riscos;
- 11.4 Realizar a análise quantitativa dos riscos;
- 11.5 Planejar as respostas aos riscos;
- 11.6 Controlar os riscos;

13. GERENCIAMENTO DAS PARTES INTERESSADAS

- 13.1 Identificar as Partes Interessadas;
- 13.2 Planejar o gerenciamento das Partes Interessadas;
- 13.3 Gestão do Engajamento das Partes Interessadas;
- 13.4 Controlar o Engajamento das Partes Interessadas;

Acessar o Processual de Gerenciamento de Projetos - PMBOK®

Baseado no PMBOK® 5th Edition (versão em Inglês).
Copyright: Project Management Institute - Um Guia do Conhecimento em Gerenciamento de Projetos - Quinta Edição (Guia PMBOK®) Project Management Institute, Inc., (2012).



Salvo mais sólido framework
de Gerenciamento de Projetos
em www.pmbok.com.br

Copyright a todos os direitos reservados.
Equipes de Trabalho:
- Geral: Comitê - Desenvolvimento;
- Técnico: Controle - Planejamento;

Note: Especificativa: trata-se de uma livre tradução e
interpretação dos processos.

Áreas de conhecimento

- **Integração** – descreve os processos que integram elementos do gerenciamento de projetos, que são identificados, definidos, combinados, unificados e coordenados dentro dos grupos de processos de gerenciamento de projetos.
- **Escopo** – processos envolvidos na verificação de que o projeto inclui todo o trabalho necessário e apenas o trabalho necessário, para que seja concluído com sucesso.
- **Tempo** – processos relativos ao término do projeto no prazo correto. Os cinco primeiros processos são de planejamento e apenas o último é de controle.
- **Custo** – processos envolvidos em planejamento, estimativa, orçamentação e controle de custos, de modo que o projeto termine dentro do orçamento aprovado.

Áreas de conhecimento

- **Qualidade** – processos envolvidos na garantia de que o projeto irá satisfazer os objetivos para os quais foi realizado.
- **Recursos Humanos** - processos que organizam e gerenciam a equipe do projeto.
- **Comunicações** - processos relativos à geração, coleta, disseminação, armazenamento e destinação final das informações do projeto de forma oportuna e adequada.
- **Riscos** - processos relativos à realização do gerenciamento de riscos em um projeto.
- **Aquisições** - processos que compram ou adquirem produtos, serviços ou resultados, além dos processos de gerenciamento de contratos.

PMBOK® Sexta Edição

Os 5 Grupos de Processos de Gerenciamento de Projetos

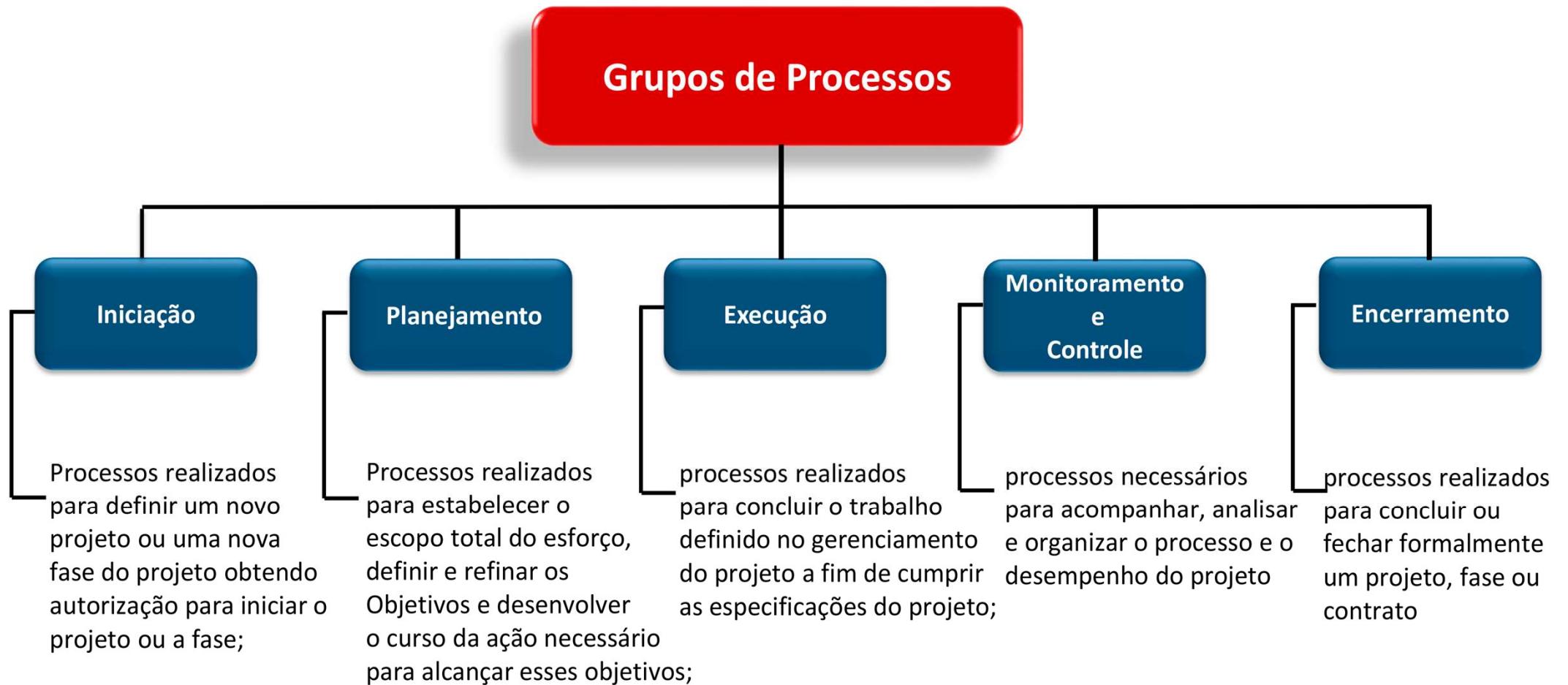


Figura baseada no PMBOK® - Sexta Edição

Aprenda +

- Vídeo "**PMBOK® Guide 6a Edição Explicado com Ricardo Vargas**" disponível em: https://www.youtube.com/watch?v=rvDnS_wWwJs
- LIMA, Fabiana. **Gerenciamento do tempo em projetos de software usando PMBOK**. Disponível em:
<https://www.devmedia.com.br/gerenciamento-do-tempo-em-projetos-desoftware-usando-pmbok/32920>
- MONTES, Eduardo. **Gerenciamento dos riscos: O que é, objetivo e processos**. Disponível em:
<https://escritoriodeprojetos.com.br/gerenciamento-dos-riscos-do-projeto>

Próxima aula – Leitura Específica

- Vídeo "Requisito Funcional e Não Funcional de Software: entenda a diferença." <https://www.youtube.com/watch?v=YLd6AWKVyas>
- SOMMERVILLE, Ian. Engenharia de Software. 10^a Ed. São Paulo: Pearson Prentice Hall, 2011. Páginas 85 até 110. Disponível em: <https://plataforma.bvirtual.com.br/Leitor/Loader/168127/pdf>
- Vídeo "Esqueça isso e seu projeto estará condenado. (Análise de Requisitos)". Disponível em:
<https://www.youtube.com/watch?v=rVbJ7ykuLig>