

Projetos de *Software*

(aula 1)

Revisão de conceitos e
introdução à projetos de
software

Me. Diogo Tavares da Silva
contato: *tavareko@gmail.com*

Antes de mais nada...

- Qual a principal função social de uma empresa?
 - Lucro?
 - criar produtos?
 - oferecer serviços?
- A resposta é:

Antes de mais nada...

- Qual a principal função social de uma empresa?
 - Lucro?
 - criar produtos?
 - oferecer serviços?
- A resposta é:

NÃO

Antes de mais nada...

- A função de uma empresa é **satisfazer as necessidades dos clientes**
 - Oferecer **SOLUÇÕES!**
- Desenvolver produtos e serviços de utilidade e eventualmente obter lucro se dá por uma soma de fatores.
 - Gerência de aspectos como tempo, custos, riscos, etc.

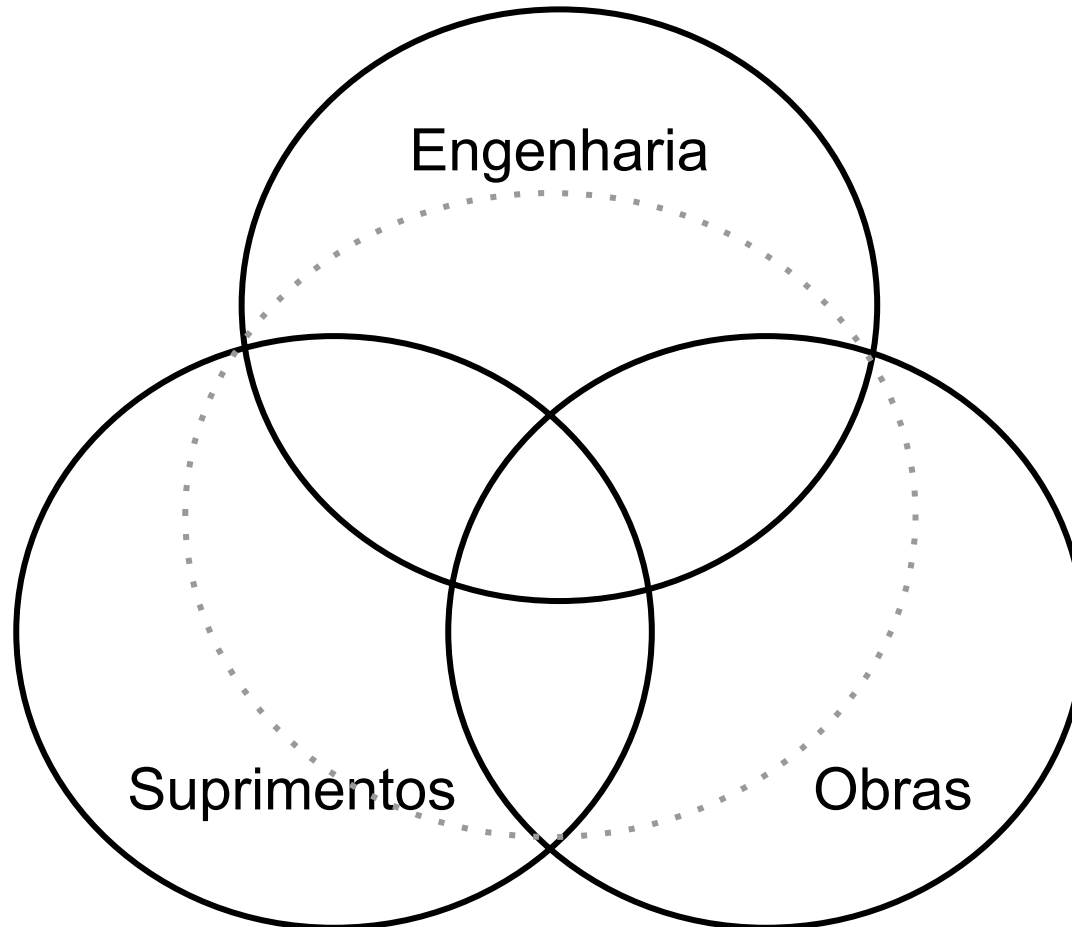
O que é um projeto?

- Um projeto é um empreendimento
 - trabalho que visa a criação de um produto ou serviço que satisfaça a necessidade de um cliente e que envolve um certo grau de incerteza na realização

Eficácia de um projeto

- Planejamento e gerência de três áreas que compõe um projeto
 - **Engenharia:** Área responsável pelas ações de especificação e modelagem do projeto
 - **Suprimentos:** Área responsável pelas ações de aquisição de materiais e recursos humanos para o desenvolvimento do projeto
 - **Obras:** Área responsável pelas ações de criação e desenvolvimento do produto ou serviço

Eficácia de um projeto



Gerência de projetos

- Área de estudo responsável por gerenciar as áreas que compõem um projeto de modo a analisar, supervisionar e coordenar as atividades de desenvolvimento do projeto de modo a controlar os fatores essenciais para que um projeto seja executado com sucesso.
- Início na década de 60
 - Indústria bélica e aeroespacial

Gerência de projetos

- **PMI - *Project Management Institute***

- Entidade internacional sem fins lucrativos
- Congrega profissionais de áreas relacionadas à disciplina de Gerência de projetos
- Missão: Promover o profissionalismo e desenvolver o “estado-da-arte” da disciplina de gerência de projetos

Gerência de projetos

- **PMBOK - *Project Management Body of Knowledge***
 - Material de referência desenvolvido pelo PMI que apresenta um conjunto de conhecimentos que visa padronizar a teoria de gerência de processos
 - identificar e descrever conceitos e práticas de gerência de projetos, padronizando a terminologia e processos utilizados

Stakeholders

- Parte mais fundamental do desenvolvimento de um projeto
- O que são?
 - Todas as pessoas que estão envolvidas direta e indiretamente na produção de um projeto
- Entender as características dos indivíduos envolvidos e seus papéis dentro do desenvolvimento do projeto é de importância VITAL

Principais *stakeholders*

- Quem geralmente são?
 - **Gerente de projeto:**
 - Responsável pela gestão, planejamento e controle do projeto
 - Gerencia a equipe de desenvolvimento
 - Responsável e comunicativo
 - Saber lidar com pressões, custos e prazos

Principais *stakeholders*

- Quem geralmente são?
 - **Apoiadores ou patrocinadores:**
 - Pessoa responsável pela administração dos projetos da empresa que desenvolve os projetos.
 - Tem como papel tomar as decisões finais sobre o projeto, apoiando seu desenvolvimento e protegendo o mesmo das influências externas e internas.

Principais *stakeholders*

- Quem geralmente são?
 - **Equipe de desenvolvimento:**
 - Pessoas envolvidas diretamente na concepção do projeto
 - Analistas, projetistas, desenvolvedores, etc
 - **Clientes:**
 - Contribui com verbas e define quais os requisitos necessários para o produto ou serviço solicitado

Principais *stakeholders*

- Quem geralmente são?
 - **Gerentes funcionais:**
 - Encarregados por gerenciar a mão de obra para a execução dos trabalhos, dar assistência especializada e alertar o gerente de projeto sobre possíveis influências na execução do projeto
 - **Outros interessados:**
 - Áreas da empresa, setores da sociedade, sindicatos, mídia, pessoas influenciadas indiretamente pelo projeto

Análise de *stakeholders*

- *Stakeholders* podem ser divididos de acordo com a influência que possuem em um projeto
 - ***Stakeholders* primários:** Estão diretamente envolvidos com o projeto
 - Empresa contratante, clientes e usuários do produto ou serviço a ser construído, equipe de desenvolvimento do projeto, etc.

Análise de *stakeholders*

- *Stakeholders* podem ser divididos de acordo com a influência que possuem em um projeto
 - ***Stakeholders* secundários:** Afetam o desenvolvimento de um projeto de forma indireta como parceiros ou adversários de um projeto
 - Ações judiciais
 - Sindicatos
 - Comunidades locais
 - Apoio da mídia

Ciclo de Vida de Projeto

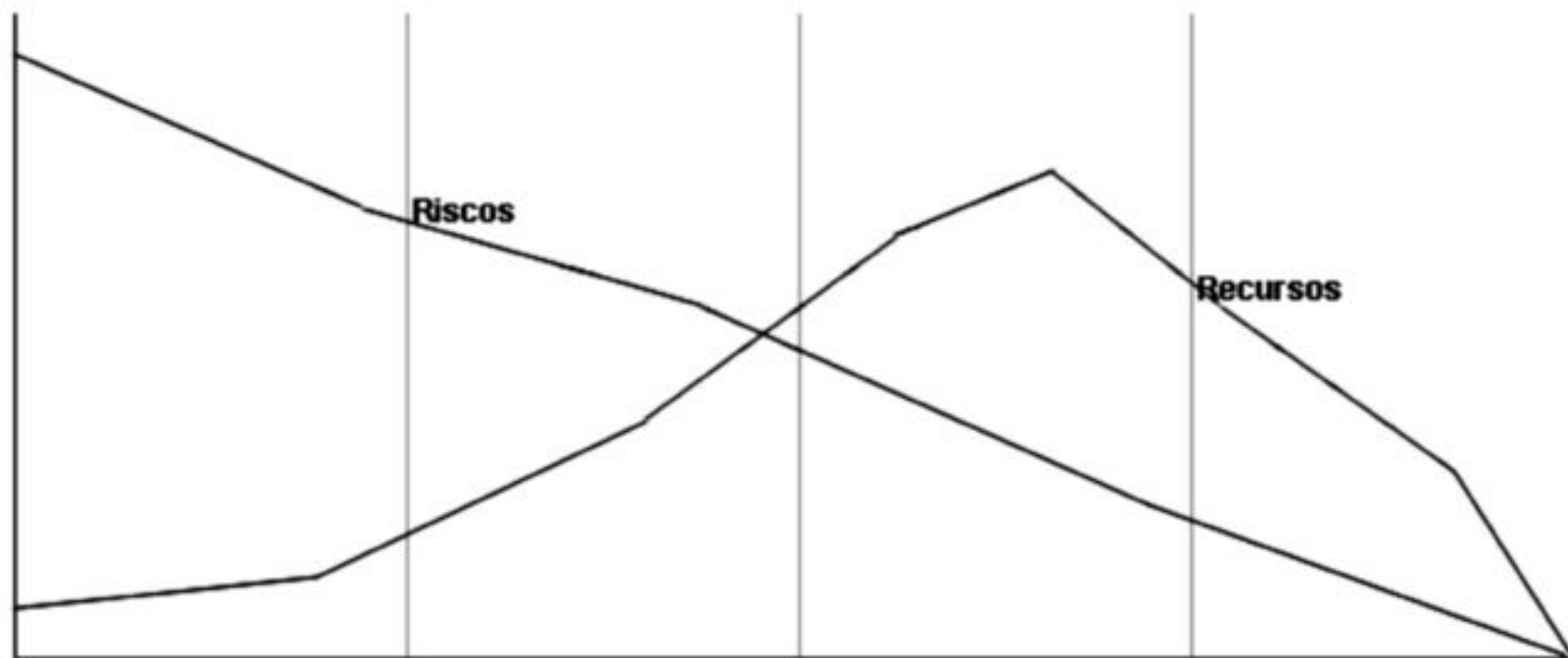
- Projetos geralmente são divididos em uma série de fases ou etapas
 - Objetivo de simplificar o controle gerencial e permitindo coordenar melhor as atividades realizadas pelos processos gerenciais.
- O conjunto de etapas que constitui um projeto é chamado de **ciclo de vida do projeto**

Ciclo de Vida de Projeto

- PMBOK não define um número de fases específico para o ciclo de vida de um projeto
- Número definido por questões de projeto
 - tamanho do projeto
 - número de pessoas envolvidas
 - complexidade do projeto
 - ciclos padrões sugeridos ou impostos pela empresa ou gerente de projeto

Ciclo de Vida de Projeto

- Segundo o PMBOK, ciclos de vida são importantes para definir:
 - Quais trabalhos técnicos serão realizados em cada fase
 - Prazos de verificação, validação, revisão e entrega de subprodutos ou produtos.
 - Pessoas envolvidas em cada fase
 - Mecanismos de controle e aprovação para cada fase
- Geralmente um ciclo de projeto encerra com a revisão do trabalho e aceitação das entregas



Conceitual

Desejos
Oportunidades
Necessidades
Metas
Objetivos
Exigências
Objeto/Escopo

Desenvolvimento

Especificação
Qualidade/Performance
WBS
Estrutura Organizacional
Responsabilidades
Plano/Estratégia
Programação
Contratações
Fluxo de Caixa
Plano de Comunicação

Realização

Plano Executivo
Materialização
Verificação da performance
(Prazo, Qualidade e Custos)
Análise da performance
Ações Corretivas

Finalização

Entrega/Operação
Arquivamento de docs.

Ciclo de Vida de Projeto

- As etapas de um ciclo de vida de projeto não precisam necessariamente ocorrer de forma progressiva
- Gerentes de projeto podem optar por realizarem etapas em paralelo
- Outra abordagem (comum em projetos de *software*) é executar os processos de forma iterativa (ciclos repetem-se de forma incremental)

Processos de gerenciamento de projetos

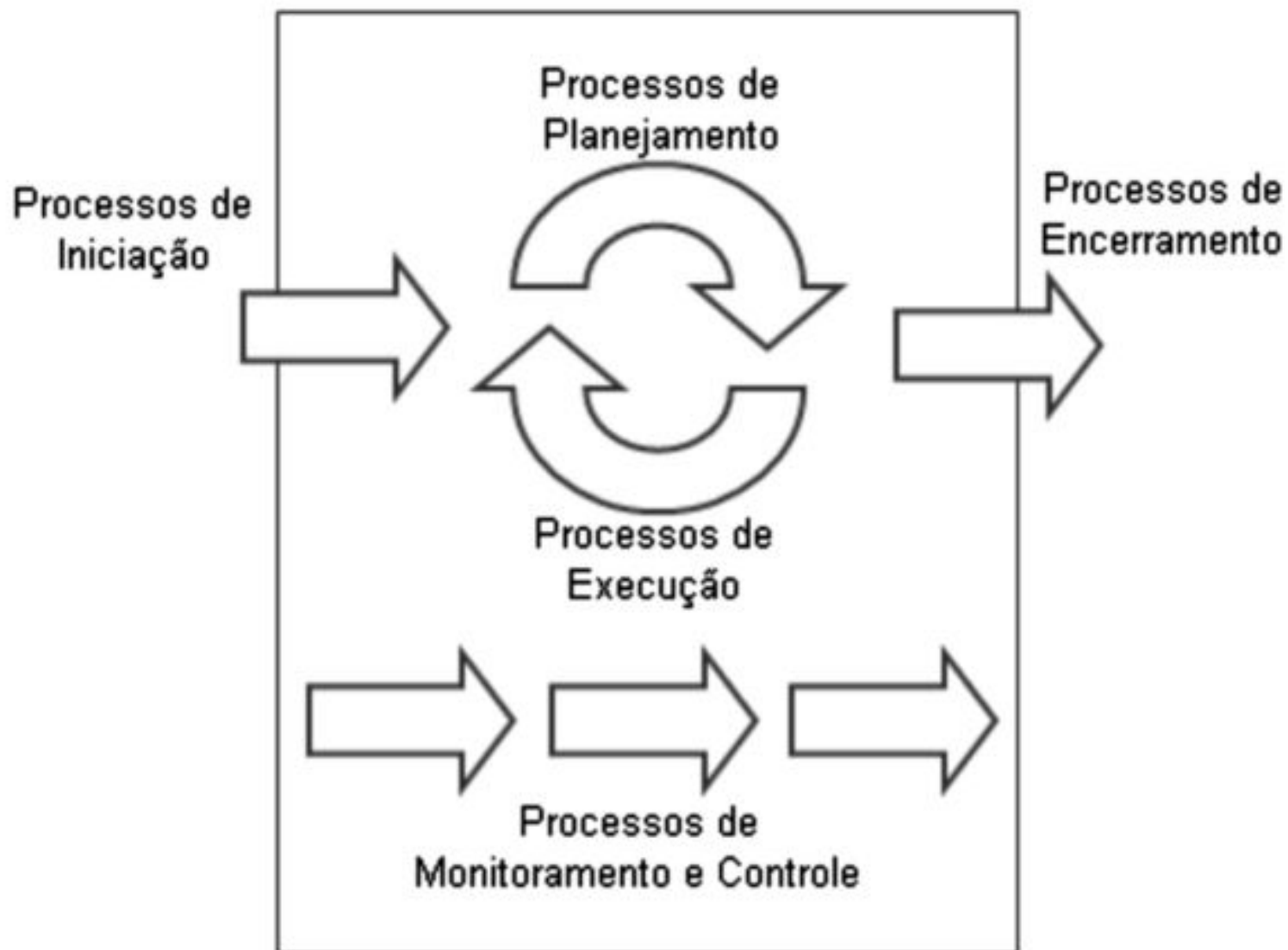
- Segundo o PMBOK, a gerência de projetos deve ser realizada por um conjunto de processos bem definidos
 - **Processos de gerenciamento**
- Divididos em cinco grupos, segundo a natureza do processo dentro do ciclo de vida do projeto.

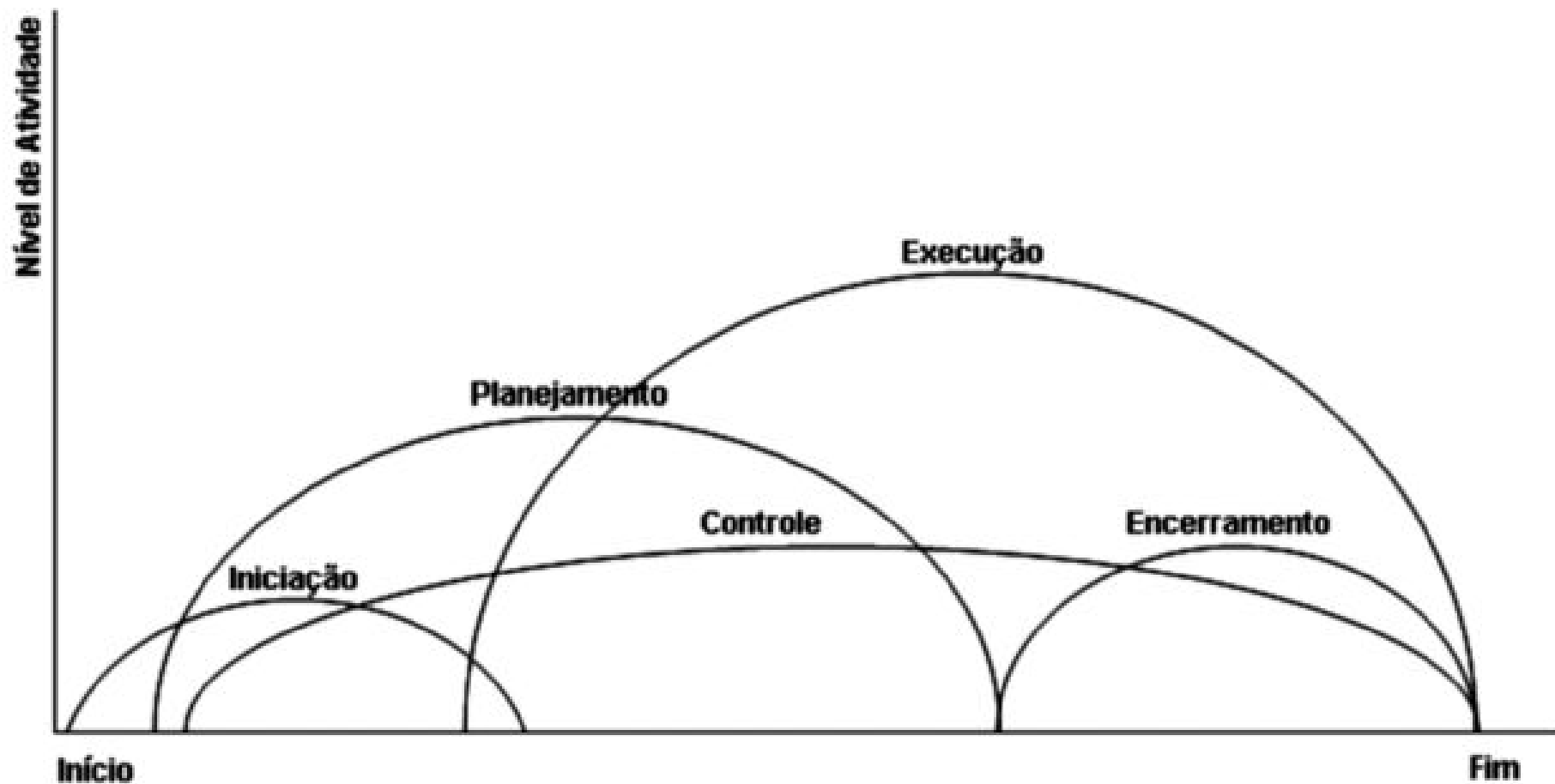
Grupos de processos de gerenciamento

- **Processos de iniciação:** Define e autoriza um projeto ou fase do projeto
- **Processos de Planejamento:** Define e refina os objetivos, o planejamento e a estratégia de implementação, assim como a programação das atividades, prazos, custos, riscos e formação de equipe.
- **Processos de execução:** Coordena as pessoas e recursos para execução do plano de projeto

Grupos de processos de gerenciamento

- **Processos de monitoramento e controle:** Mede e monitora regularmente o progresso do projeto visando identificar desvios em relação ao planejamento, de forma que possam ser implementadas ações corretivas para trazer o projeto de volta ao seu caminho planejado.
- **Processos de finalização:** Formaliza a aceitação dos produtos e serviços entregues, ou qualquer outro resultado importante do projeto de uma fase





PMBOK vs Metodologia de projeto

- PMBOK NÃO é uma metodologia de desenvolvimento de projeto
 - Material de referência de BOAS PRÁTICAS de desenvolvimento de projetos.
 - Descrição de linhas gerais de projeto:
 - Ciclos de vida de projeto, análise de *stakeholders*, conjunto de processos de gerenciamento, etc.

Metodologia de projeto

- Metodologia de desenvolvimento de projeto
 - Busca estabelecer uma abordagem detalhada das atividades necessárias para desenvolver e aperfeiçoar um projeto.
 - Especifica etapas, tarefas, técnicas, linguagens e produtos necessários durante todo o ciclo de vida do projeto para seu desenvolvimento.

Metodologia de projeto

- Exemplos:
 - RUP (*Rational Unified Process*)
 - Scrum
 - XP (*Extreme Programming*)
 - Modelagem ágil
- Serão abordados com mais detalhes durante o curso

O que é *software*?

- “...programas de computador”
 - mais do que isso!
- Um *software* é um programa de computador, juntamente com seus arquivos de configuração e documentação associada
- Um sistema de software comercial, por sua vez, geralmente é ainda mais complexo.
 - Ecossistema de *software*, composto por vários programas que geralmente possuem conectividade e integração de dados

Importância do *software*

- Um *software* é um veículo de entrega e transformação do produto mais importante dos dias atuais: a INFORMAÇÃO
- Toda economia global atual está completamente dependente do uso de sistemas de informação
 - grande desenvolvimento de empresas que produzem *software* como um produto comercial
 - Volume crescente de informações criam novas oportunidades de negócio (Big Data, Machine Learning IoT etc)

Software* amador X Produto de *software

- Software de uso amador possui desenvolvimento menos criterioso quanto a codificação e documentação
- Um produto de *software* por sua vez deve ser produzido seguindo uma série de critérios
 - deve haver documentação de apoio ao projeto e de apoio ao usuário
 - existem obrigações contratuais com clientes
 - custos e prazos para serem seguidos
 - responsabilidade de manutenção e suporte

Projetos de *software*

- Um bom produto de software, assim como qualquer outro bom produto, é consequência de um fator em comum:
 - O desenvolvimento através de um **bom projeto**
 - Um **boa gerência de projeto** é essencial para um resultado final positivo
 - No entanto, o desenvolvimento de *software* possui **peculiaridades**

Projetos de *software*

- Diferente de um produto ou serviço físico, um produto de *software* é abstrato e intangível
 - Não está limitado a condições físicas de materiais e processos de manufatura
 - Não existem limitações físicas para um produto de *software*
 - Um produto de *software* deve estar adaptado ao ambiente, sofrendo alterações sempre que o ambiente mudar

Projetos de *software*

- O aspecto intangível do *software* acaba levando a uma série de questões como:
 - Por que concluir um *software* leva tanto tempo?
 - Porque os custos de desenvolvimento são tão altos?
 - Porque não conseguimos encontrar todos os erros antes de entregarmos o *software* os clientes?

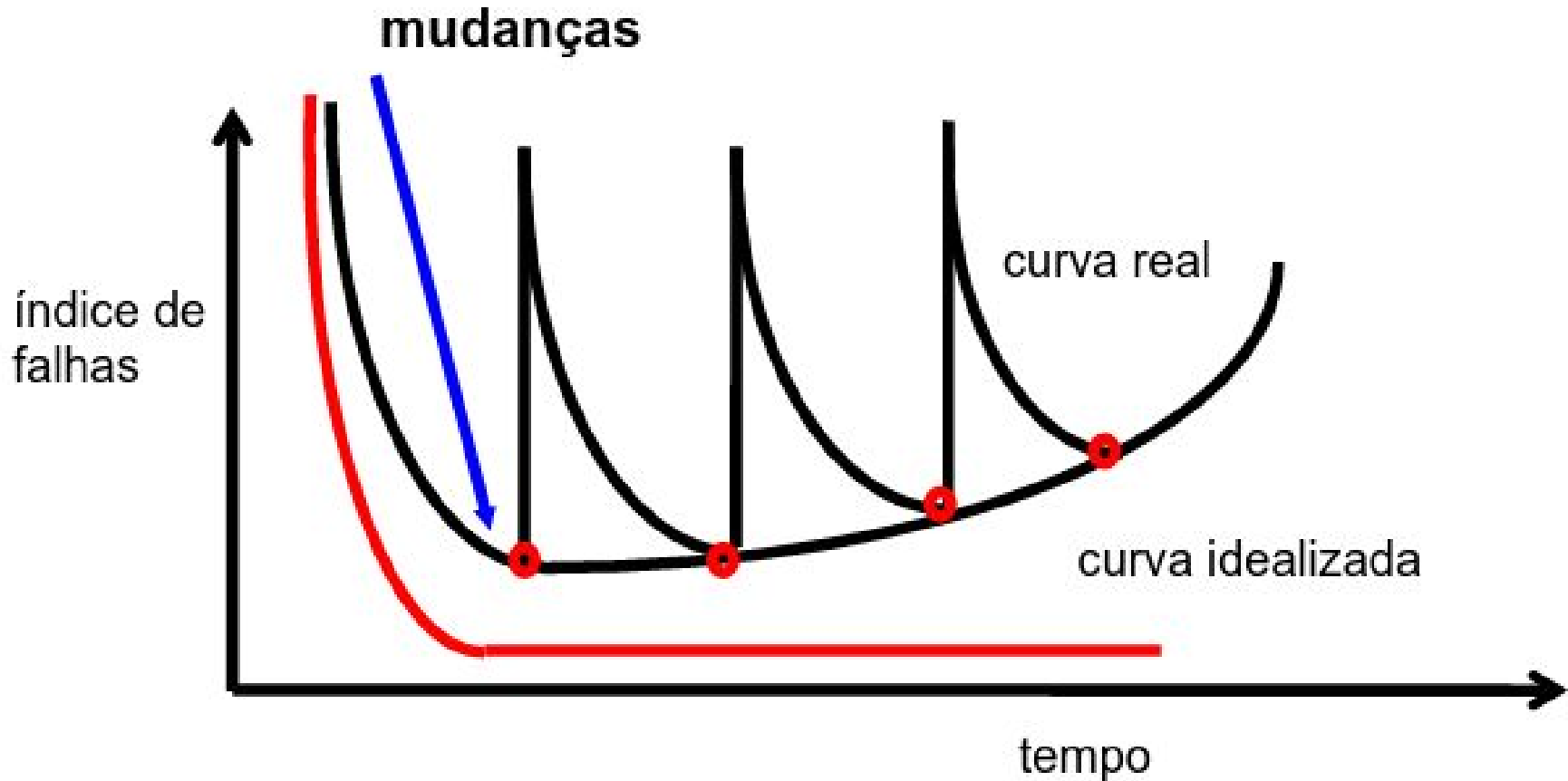
Projetos de *software*

- Porque gastamos tanto tempo e esforço mantendo programas existentes?
- Porque continuamos a ter dificuldade em medir o progresso enquanto o *software* está sendo desenvolvido e mantido?

Curva de defeitos para *hardware*



Curva de defeitos para *software*



Software não se desgasta, mas se deteriora devido às mudanças!

Por que o software precisa mudar?

- Existem três fatores que têm se mostrado como os principais influenciadores no modo como software é projetado e desenvolvido:
 - **Heterogeneidade:** Aumento da demanda para desenvolvimento em sistemas computacionais variados (*desktop, websites, mobile devices, CDN, cloud computing, S.O.A.*).

Por que o software precisa mudar?

- **Mudanças ambientais (lógicas de negócio, legislações e relações sociais):** Caso o ambiente em que o *software* está inserido muda o mesmo deve mudar:
 - Alterações das lógicas de negócio ou cadeias produtivas
 - Alteração das legislações vigentes
 - Integração de funcionalidades relacionadas às interações sociais (*social networking*)

Por que o software precisa mudar?

- **Segurança e confiabilidade:**

- Descobrimiento de falhas ou vulnerabilidades
- Aumento do número de informações pessoais e da integração entre essas informações
 - necessidade de maior atenção na maneira como o *software* manipula e protege essas informações para garantir segurança dos dados e confiabilidade do uso da ferramenta

Engenharia de software

- **Gerência de projetos** é uma ferramenta importante para o desenvolvimento de um produto de *software* de qualidade, mas não é o bastante
- É necessário embasar-se em uma disciplina que aborda as peculiaridades inerentes do processo de desenvolvimento de *software*
 - Neste contexto, é que surge a **Engenharia de *software***

Engenharia de software

- Disciplina da engenharia cujo foco está em todos os aspectos da produção de *software*, desde os estágios iniciais de especificação do sistema até sua manutenção e evolução, quando o sistema já está sendo usado.
- Combina aspectos da gerência convencional de projetos juntamente com o desenvolvimento de técnicas, ferramentas e conhecimentos específicos para o desenvolvimento de *software*

Processo de software

- Assim como na gerência de projetos convencional, a engenharia de *software* deve desenvolver uma abordagem sistemática e organizada
 - **Processo de *software***
- Sequência de atividades que leva à produção de um produto de *software*

Processo de software

- Pode ser dividido em quatro etapas fundamentais:
 - **Especificação de software:** clientes e engenheiros definem como deve ser o *software* a ser produzido e quais as restrições de sua operação
 - **Desenvolvimento de software:** Etapa em que o software é projetado e programado

Processo de software

- Composto basicamente por quatro atividades fundamentais
 - **Validação de software:** Em que o software é testado, verificado e validado para garantir que está como o cliente deseja
 - **Evolução de software:** Em que o software é modificado para refletir a mudança de requisitos do cliente e do mercado

Processo de software

- Não existe um único modo de se conduzir um *processo de software*.
 - Sistemas de software podem ser de várias naturezas
 - processos de *software* mais adequados para cada tipo de software.
- Conjunto comum de atividades de desenvolvimento, porém organizados e estruturados de maneiras distintas de acordo com o produto a ser desenvolvido
 - ***assunto para próxima aula**



Variedade de sistemas computacionais

- Sistemas *stand-alone*
- Sistemas transacionais
- Sistemas de controle embutido
- Sistemas de processamento em lotes
- Sistemas de entretenimento
- Sistemas de modelagem e simulação
- Sistemas de coletas de dados
- Sistema de sistemas
- entre outros

Ferramentas CASE

- ***Computer-Aided Software Engineering***
 - Ferramentas de *software* que auxiliam no desenvolvimento de atividades de engenharia de *software*
 - Ferramentas de análise, prototipação, modelagem, desenvolvimento, testagem, etc.
 - Conheceremos várias delas no decorrer do curso