

FIA P GRADUAÇÃO

ASSUNTO:

ENGENHARIA E DESIGN DE SOFTWARE

AULA:

1-INTRODUÇÃO À MODELAGEM DE SISTEMAS DE INFORMAÇÃO

PROFESSOR:

RENATO JARDIM PARDUCCI

PROFRENATO.PARDUCCI@FIAP.COM.BR



APRESENTAÇÃO DO PROFESSOR

I SHORT BIO



RENATO JARDIM PARDUCCI

Profrenato.parducci@fiap.com.br

Apresentações

Prof. Renato Jardim Parducci



Astro nas horas extras



CREATED USING
BwToon

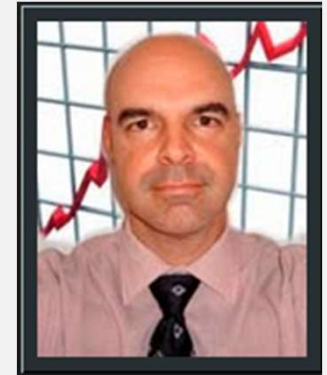


PREPARAÇÃO PARA MINISTRAR A DISCIPLINA

- **Membro da equipe de desenvolvimento de sistemas e teste de software da CETIL, Leão Engenharia e BG&C.**
- **Líder de desenvolvimento de sistemas responsável pela implantação de metodologia de modelagem na Itaotec, Philco.**
- **Gestor da área de produção de software e bancos de dados na OESP-Bell South e no grupo O Estado de S. Paulo.**
- **Gestor de infraestrutura e serviços de TI, Redes e Telecomunicações e administração de Bancos de Dados na Leão Engenharia, BG&C, OESP-Bell South, Estadão, EMS-Pharma**
- **Consultor e gestor de implantação de programas de Qualidade de software, Fábrica de software, Escritório de projetos, Centrais de operações de TI e Suporte técnico, programas de Governança, Planejamento e estruturação estratégica de TI – Netpartners, Grupo Linx, Animatech, EMS-Pharma**
- **Gerente de TI nas empresas Unilever, OESP Mídia e Gráfica, Estadão, Rádio Eldorado, Agência Estado, ,EMS Pharma, Legrand, Germed, Nova Química, ACS incorporadora.**
Diretor de TI nas empresas Netpartners e LINX.

Diretor sócio nas empresas RUNAK Tecnologia e RUNMídia

Formação



Bacharel em Ciência da Computação



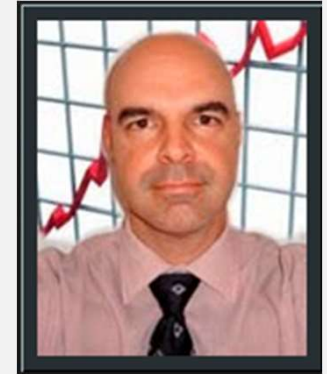
Especialista em Adm. Industrial



Mestre em Eng. Computação

Muita experiência para trocar!

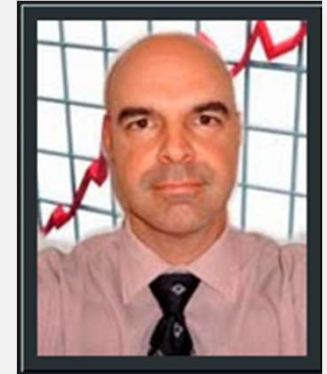
Analista de negócios, de sistemas e desenvolvedor de software:



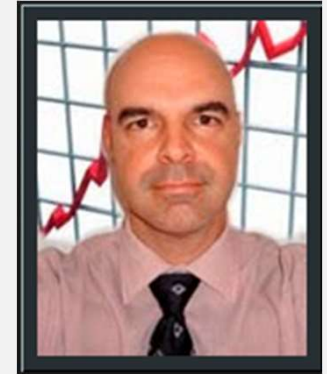
Executivo da área de TI:



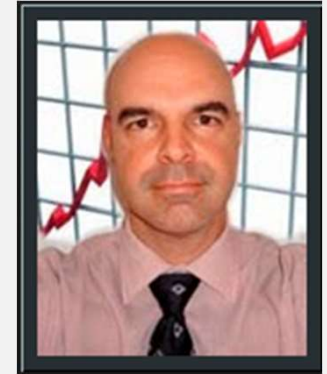
Consultor em projetos de diversas empresas renomadas



Mentor de Startups

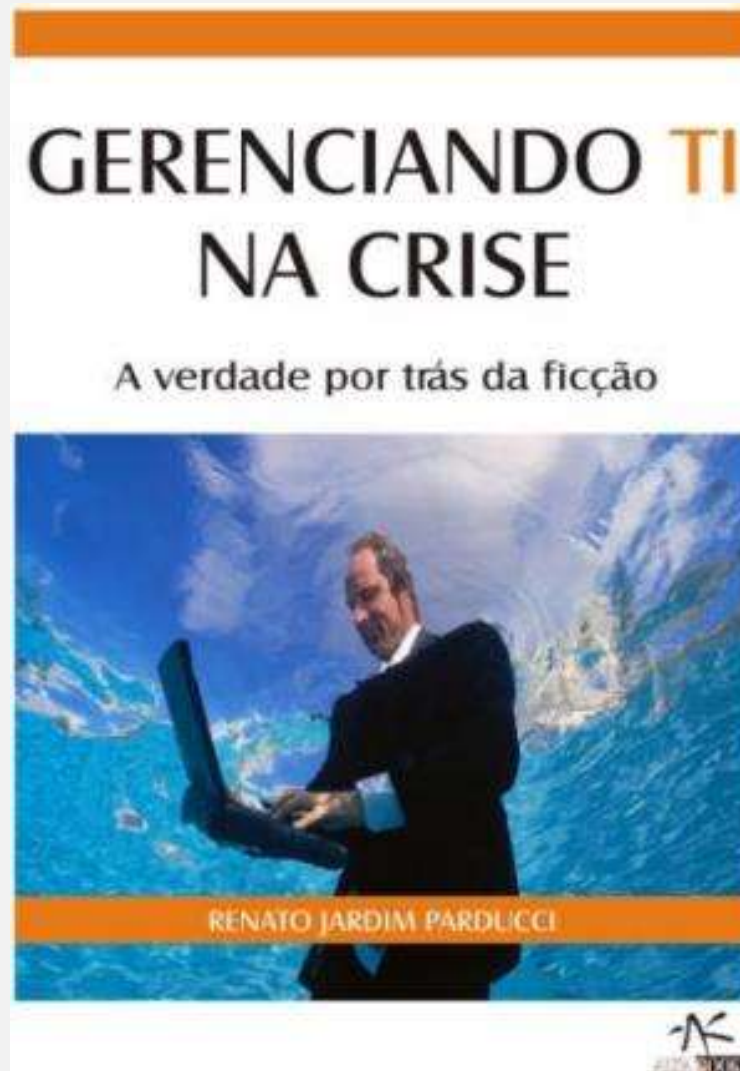


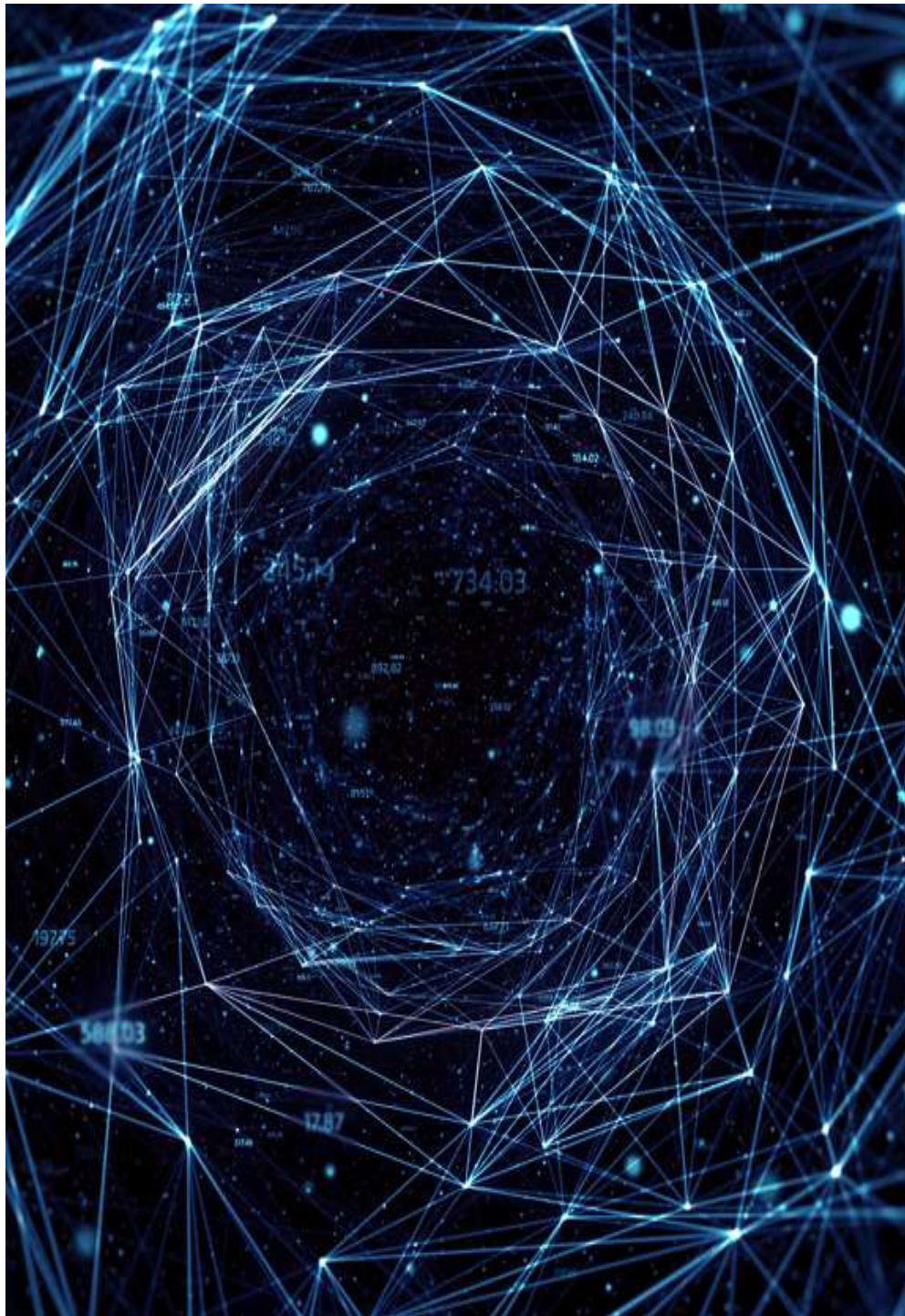
Atividade acadêmica intensa





PUBLICAÇÕES





Apresentação da Disciplina

AGILE SOFTWARE DESIGN

ESCOPO DA DISCIPLINA

PROJETOS ÁGEIS DE SOFTWARE

- .Engenharia de software – princípios
- .Ciclos de vida, processos de software e o paradigma ágil
- .Engenharia de requisitos em processo ágil
- .Planejamento, execução, acompanhamento e controle de projeto ágil

DESENHO TÉCNICO DE SOFTWARE

- .Padrões para especificar software
- .UML aplicada à modelagem de software
- .Codificação de software baseada no design
- .Ferramentas CASE

AGILE SOFTWARE DESIGN

NOSSA META

Competências	<p>Ao final do curso o aluno será capaz de:</p> <ul style="list-style-type: none">▪ Identificar e selecionar os requisitos que o software deve atender;▪ Especificar o sistema de informação a ser construído, detalhando sua arquitetura e funcionalidades através de uma linguagem de modelagem de sistemas – a UML;▪ Planejar, acompanhar e monitorar o desenvolvimento de software de forma ágil▪ Gerenciar os produtos de software gerados.
Habilidades	<p>Analisar e interpretar desafios de desenvolvimento de software aplicando raciocínio lógico; planejamento, organização, controle e comunicação de projeto com condução ágil.</p> <p>Desenvolver a engenharia do software aplicando técnicas de::</p> <ul style="list-style-type: none">▪ Levantamento, registro, análise de viabilidade e priorização de Requisitos▪ Diagramação e descrição de casos de uso▪ Diagramação de classe▪ Diagramação de atividades▪ Diagramação de sequencia▪ Diagramação de estados▪ Diagramação de componentes▪ Diagramação de pacotes▪ Diagramação de modelos de implementação▪ Uso de ferramenta CASE▪ Uso de plataforma integrada de gestão ágil e documentação de projeto de software

AGILE SOFTWARE DESIGN

DETALHES DO CONTEÚDO

1º Semestre	2º Semestre
<p>Unidade I – Ciclo de vida, processo de software e modelagem de sistemas com métodos ágeis</p> <ul style="list-style-type: none"> • Atividade do profissional <i>full stack</i> na visão da arquitetura empresarial - TOGAF-ADM • Ciclos de vida de software (Cascata, Incremental, Prototipação e evolucionária e Espiral) • Processos tradicionais e ágeis de software • Modelagem de sistemas inserida nos processos e ciclos de vida • Visão geral do SCRUM – processo, papéis e responsabilidades (PO, Scrum master, time de desenvolvimento) <p>Unidade II – Engenharia de requisitos</p> <ul style="list-style-type: none"> • Conceito de Análise e Projeto Lógico e Físico • Requisito de software (Requisitos funcionais e não funcionais, Requisitos de usuários, Requisitos de sistema, Requisitos de domínio) • Construção do backlog de produto no SCRUM – Temas, Épicos, Histórias de usuário e o critério de pronto • Técnicas de Levantamento de Requisitos (pesquisa, questionário, entrevista, workshop, observação de campo, JAD) • Análise de viabilidade de requisitos • Priorização de requisitos <p>Unidade III – Planejamento de projeto ágil de software com Scrum</p> <ul style="list-style-type: none"> • Documentação de backlog de produto e requisitos no AZURE Boards • Definição de plano de release de produto • Sprint <i>plan</i>, com abertura das tarefas de modelagem, construção e testes <p>Unidade IV – Engenharia de software orientado a objetos com UML</p> <ul style="list-style-type: none"> • Visão geral da UML e seus modelos dinâmicos e estáticos • Diagramas de Casos de Uso • Documentação de Caso de Uso • Cenários de Uso 	<p>Unidade II – Engenharia de software orientado a objetos com UML (continuação)</p> <ul style="list-style-type: none"> • Diagramas de atividades • O paradigma orientado a objetos e seus principais mecanismos • Diferenças entre o desenvolvimento procedural e a O.O. • Modelagem de Objetos (Identificação de Objetos e Classes candidatas, com base nos Casos de Uso) • Modelagem de Classe (responsabilidade de Classe, declaração de estruturas, Identificação, Atributos e Métodos, especialização e generalização de Classes, associação de Classes, <i>componentização</i>, agregação, classes Associativas, herança e polimorfismo, visibilidade de atributos e <i>métodos</i>). • Especificação <i>Algorítmica</i> de métodos • Arquitetura MVC (<i>Model View Control</i>) com Classes estereotipadas. • Projeto físico e construção de software com base no modelo de Classes • Modelagem da Comunicação entre Objetos e Classes • Diagramação de Atividade (aplicado aos Métodos) • Diagramação de Sequência • Diagramação de Estados • Diagramação de Componentes • Diagramação de Pacotes • Diagramação de Implementação • CASE aplicado (Modelagem de software assistida por computador, geração de código fonte a partir da especificação de Classes de objetos com ASTAH)

AGILE SOFTWARE DESIGN

MATERIAL DE AULA DE APOIO

SLIDES DE AULA

.Material didático com teoria e exemplificação prática

ESTUDOS DE CASO

.Material contendo desafios e soluções realizados em sala de aula, com base em estudo de caso prático

EXERCÍCIOS

.Lista de exercícios resolvidos (com resposta no próprio material) e propostos (desafios para o estudante que não têm resposta no material)

JOGOS DE FIXAÇÃO

CANAL DO PROFESSOR

.Vídeos sobre ferramentas e métodos estudados, com link nos slides de aula

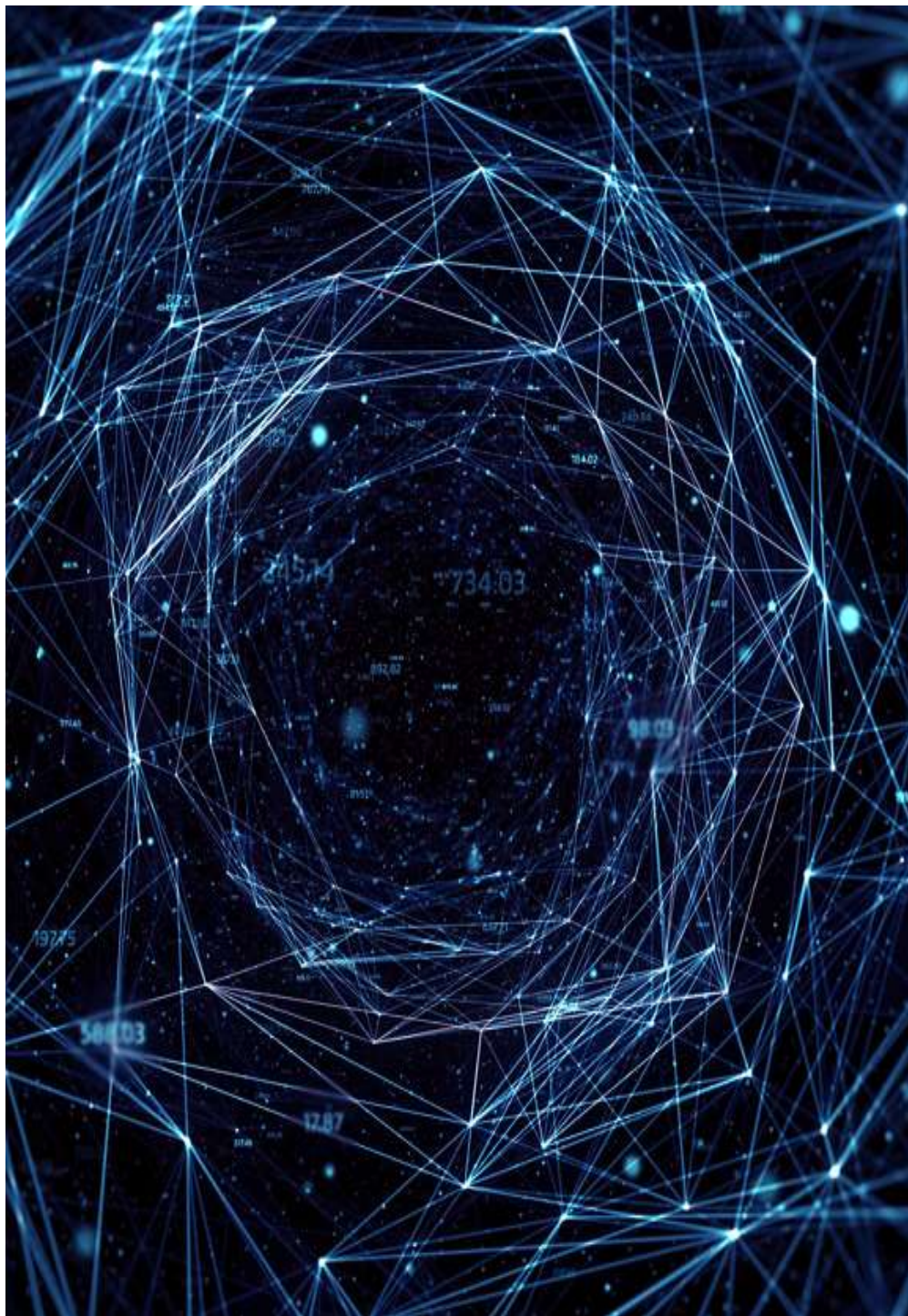
AGILE SOFTWARE DESIGN

BIBLIOGRAFIA

Bibliografia	
Básica	<ol style="list-style-type: none">1- SOMMERVILLE, I. Engenharia de software. 8ª ed. São Paulo: Pearson Education do Brasil, 2010.2- LIMA, Adilson da Silva. UML 2.5 do Requisito a Solução. 1ª ed. São Paulo: Editora Erica, 2016.3- Rubin, Kenneth S. Scrum Essencial - Um Guia Prático Para o Mais Popular Processo Ágil. Alta Books. 2018.
Complementar	<ol style="list-style-type: none">1- LAUDON, K. C.; Laudon J. P. Sistemas de Informação Gerenciais. 11ª ed. São Paulo: Pearson Education do Brasil, 2015. *2- PFLEEGER, Shari Lawrence. Engenharia de software: teoria e prática. 2ª ed. São Paulo: Pearson Education do Brasil, 2004. *3- PAGE-JONES, Meilir. Fundamentos do desenho orientado a objetos com UML. 1ª ed. São Paulo: Pearson Education do Brasil, 2001. *4- VAZQUES, Carlos Eduardo; SIMÕES, Guilherme Siqueira. Engenharia de Requisitos. São Paulo: Brasport, 2016. *5- MEDEIROS, E. S. Desenvolvendo Software com UML 2.0: definitive. 1ª ed. São Paulo: Pearson Education do Brasil, 2004. *6- GUEDES, G. T. A. UML2 – Uma abordagem prática. 2ª ed. São Paulo: Novatec, 2011.

(*) Acervo online

Outras bibliografias poderão ser indicadas nos materiais de apoio



VAMOS
COMEÇAR!

AGENDA DA AULA

VISÃO GERAL DAS RESPONSABILIDADES SOBRE O DESENVOLVIMENTO DE SOFTWARE:

- ✓ Profissionais
- ✓ Funções
- ✓ Responsabilidades

INTRODUÇÃO AO CICLO DE VIDA DE UM SOFTWARE:

- ✓ Fases de desenvolvimento
- ✓ Durabilidade de software

ATIVOS INFORMACIONAIS

APRENDENDO NA PRÁTICA



Você atua em uma software house startup que constituiu com alguns amigos da faculdade.

Como seu primeiro projeto, você vai desenvolver em sua empresa um sistema de informação para um proprietário de estacionamento, conhecido de um dos seus sócios.

Como o estacionamento é todo controlado manualmente, o proprietário quer entender um pouco mais sobre como funciona “esta história de computação” nos negócios.

Sua primeira tarefa será esclarecer a esse proprietário, o que compõe um projeto de sistema e você preparou a apresentação a seguir, a qual você deve revisar.

Visão dos elementos envolvidos em um projeto de Tecnologia



HARDWARE

A parte “dura” da informática.

Composta pelos equipamentos (computadores, equipamentos de rede, cabeamento).

Sem flexibilidade/facilidade para ser modificado.

Uma modificação geralmente implica em criar um novo produto.



SOFTWARE

A parte “mole” da informática.

Composta pelos programas de computador.

Bastante flexibilidade/facilidade para ser modificado.

Uma modificação pode ser feita sem criar um novo produto.



PEOPLEWARE

Profissionais da área de TI.

Aplicam suas competências e conhecimentos sobre processos de trabalho para atender usuários da tecnologia, entregando soluções para uso e auxiliando em dificuldades do dia a dia.



Visão dos elementos envolvidos em um projeto de Tecnologia



HARDWARE

A parte “dura” da informática.

Composta pelos equipamentos (computadores, equipamentos de rede, cabeamento).

Sem flexibilidade/facilidade para ser modificado.

Uma modificação geralmente implica em criar um novo produto.



SOFTWARE

Nosso foco!

A parte “mole” da informática.

Composta pelos programas de computador.

Bastante flexibilidade/facilidade para ser modificado.

Uma modificação pode ser feita sem criar um novo produto.



PEOPLEWARE

Profissionais da área de TI.

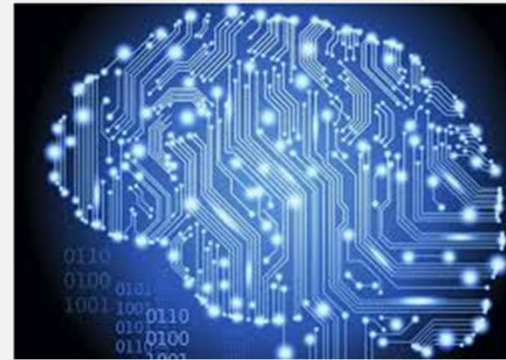
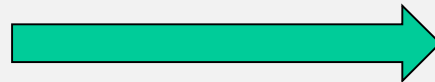
Aplicam suas competências e conhecimentos sobre processos de trabalho para atender usuários da tecnologia, entregando soluções para uso e auxiliando em dificuldades do dia a dia.

Visão dos elementos envolvidos em um projeto de Tecnologia

SOFTWARE

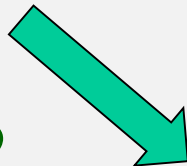


Possibilita



UTILIDADE/APLICAÇÃO
para atividades do usuário

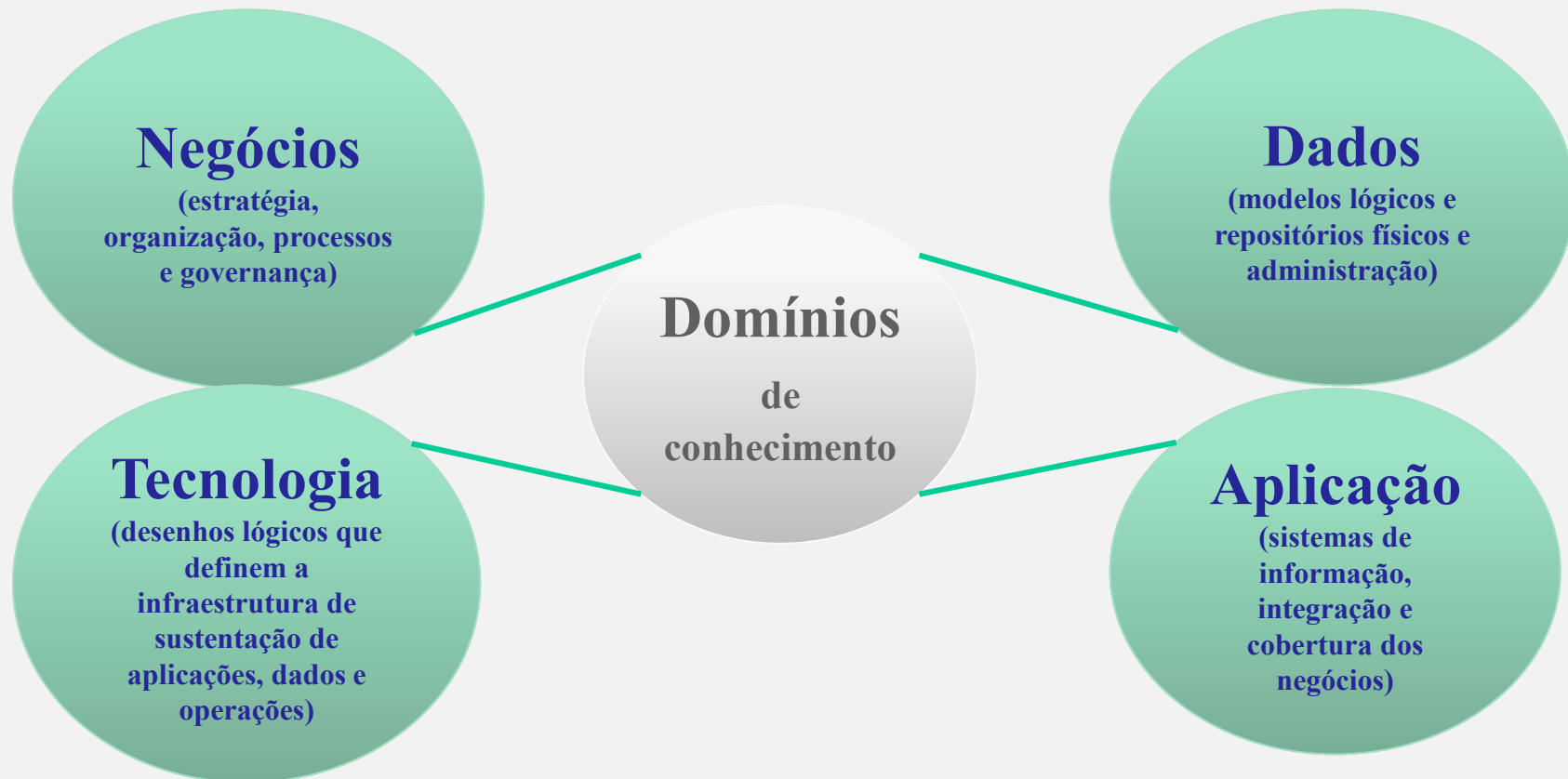
Controla o
funcionamento



HARDWARE

**FUNÇÕES DOS
PROFISSIONAIS DO
DESENVOLVIMENTO DE
SOFTWARE**

As atividades de desenvolvimento segundo o modelo de arquitetura TOGAF



APRENDENDO NA PRÁTICA



O proprietário do estacionamento ficou preocupado com a sua explicação no que tange a quantidade de pessoas que serão envolvidas no projeto.

Ele ouviu falar que gasta-se muito dinheiro em projetos de sistemas e muitas vezes, vários profissionais aparecem computando horas a pagar.

Nesse momento, você percebe que é necessário explicar ao seu cliente, quais são os perfis de profissionais que podem ser necessários de empregar durante o projeto.

Junto com os demais sócios, vocês vão relacionar os principais perfis da área de informática.

É preciso distinguir **FUNÇÕES** (ênfases de responsabilidade e competências) para atender cada objetivo de um processo de desenvolvimento de software.

Negócios

(estratégia, organização, processos e governança)

Tecnologia

(desenhos lógicos que definem a infraestrutura de sustentação de aplicações, dados e operações)



VOCÊ CONHECE?

Papéis e responsabilidades na produção de Software

Dados

(modelos lógicos e repositórios físicos e administração)

Aplicação

(sistemas de informação, integração e cobertura dos negócios)

Não é obrigação nas empresas ter que **separar cargos** para todas as funções, **porém, as FUNÇÕES** são papéis e responsabilidades que **têm que estar com alguém!**

- Não temos o objetivo de esgotar o estudo de possíveis distribuições de perfis profissionais na área de TI mas vamos conhecer os mais comuns de serem encontrados.

Resumo das funções ligadas diretamente com a produção de software



- **Engenheiro de Software**
- **Arquiteto de Solução**
- **Analista de Processos e Negócio**
- **Analista de Sistema**
- **Programador de Software**

- Define ciclo de vida e processo de software a ser seguido
- Determina padrões de documentação de projeto
- Determina padrões de codificação e testes
- Estima projetos de software em termos de esforço e custo
- Escolhe as ferramentas de apoio ao desenvolvimento de software
- Orienta e capacita a equipe de desenvolvimento
- Gerencia tecnicamente o projeto de software, geralmente em conjunto com um gerente administrativo

Resumo das funções ligadas diretamente com a produção de software



- **Engenheiro de Software**
- **Arquiteto de Solução**
- **Analista de Processos e Negócios**
- **Analista de Sistema**
- **Programador de Software**

- Estabelece padrões de separação de componentes de software, criando especialidades de serviços ou camadas de implementação
- Define protocolos de comunicação para integração de componentes
- Descreve critérios para disponibilizar e catalogar componentes reusáveis em uma biblioteca de apoio à construção de software
- Determina os critérios para criação de módulos e pacotes de produto

Resumo das funções



- **Engenheiro de Software**
- **Arquiteto de Solução**
- **Analista de Processos e Negócio**
- **Analista de Sistema**
- **Programador de Software**

- Identifica requisitos de mudança na forma de trabalho das áreas de negócio
- Avalia a prioridade dos requisitos para o negócio
- Documenta ou coleta documentos sobre fluxos de trabalho atuais e os sistemas de informação envolvidos
- Propõe mudanças em processos e sistemas
- Negocia mudanças com áreas funcionais
- Encaminha demandas para desenvolver o software
- Homologa solução com o cliente

Resumo das funções



- **Engenheiro de Software**
- **Arquiteto de Solução**
- **Analista de Processos e Negócios**
- **Analista de Sistema**
- **Programador de Software**

- Analisa viabilidade e prioridade técnica de requisitos
- Desenha o software (modela) em termos de estrutura de dados e comportamento funcional
- Elabora protótipos de interfaces
- Valida modelos contra requisitos
- Segue e alinha padrões de arquitetura de solução com Arquitetos
- Interage com programadores
- Testa o sistema integrado
- **Pode ter um desmembramento de função de Interface Designer**

Resumo das funções



- **Engenheiro de Software**
- **Arquiteto de Solução**
- **Analista de Processos e Negócio**
- **Analista de Sistema**
- **Programador de Software**

- Produz o código fonte do software em uma linguagem de programação
- Cria estruturas de dados de apoio, incluindo arquivos internos
- Segue protocolos e linguagens determinados pela Engenharia e Arquitetura
- Testa a aplicação construída
- Documenta internamente e externamente a aplicação

Resumo das funções



- **Administrador de Dados**
- **Administrador de Banco de Dados**
- **Administrador de Infraestrutura**
- **Analista de Operação**
- **Oficial de Segurança**
- **Analista de Suporte**

- Modela os bancos de dados e define padrões para estruturação de dados por desenvolvedores
- Documenta dicionários de dados
- Alinha modelos com Analistas de Sistemas
- Acompanha o uso e consumo dos dados armazenados
- Desenvolve aplicações em linguagem 4GL (declarativas de bancos de dados como SQL)
- Atua na Governança de dados

Outras funções que apoiam o desenvolvimento de software



- **Administrador de Dados**
- **Administrador de Banco de Dados**
- **Administrador de Infraestrutura**
- **Analista de Operação**
- **Oficial de Segurança**
- **Analista de Suporte**

- Instala SGBD
- Ajusta parâmetros de desempenho do banco de dados
- Cria repositórios de dados
- Cria bancos de dados
- Cria tabelas em banco de dados
- Configura acessos aos dados, respeitando regras de Segurança
- Carrega e transporta dados sob demanda do Administrador de Dados, Analista de Sistemas ou Programador

Outras funções que apoiam o desenvolvimento de software



- **Administrador de Dados**
- **Administrador de Banco de Dados**
- **Administrador de Infraestrutura**
- **Analista de Operação**
- **Oficial de Segurança**
- **Analista de Suporte**

- Instala infraestrutura servidora
- Configura acessos ao ambiente respeitando regras de Segurança
- Instala aplicações de software e utilitários
- Administra uso e consumo da infraestrutura disponibilizada

Outras funções que apoiam o desenvolvimento de software



- **Administrador de Dados**
- **Administrador de Banco de Dados**
- **Administrador de Infraestrutura**
- **Analista de Operação**
- **Oficial de Segurança**
- **Analista de Suporte**

- Executa rotinas de manutenção da infraestrutura, segurança de dados e processamento de dados em datacenter

Outras funções que apoiam o desenvolvimento de software



- **Administrador de Dados**
 - **Administrador de Banco de Dados**
 - **Administrador de Infraestrutura**
 - **Analista de Operação**
 - **Oficial de Segurança**
 - **Analista de Suporte**
- Define e divulga políticas de segurança da informação (acessos a dados e uso de devices e software)
 - Configura ou orienta a configuração de equipamentos e softwares de segurança
 - Audita o cumprimento das políticas estabelecidas

Outras funções que apoiam o desenvolvimento de software



- **Administrador de Dados**
- **Administrador de Banco de Dados**
- **Administrador de Infraestrutura**
- **Analista de Operação**
- **Oficial de Segurança**
- **Analista de Suporte**
 - Atende o usuário final
 - Soluciona incidentes da rotina de uso e operação
 - Capacita usuários
 - Interage com demais funções da TI para solucionar problemas
 - Colabora com sugestões para melhoria do software e infraestrutura

Outras funções que apoiam o desenvolvimento de software



- **Cientista de Dados**

- Mapeia o comportamento dos dados buscando tendências que auxiliam na tomada de decisão em negócios de todos os ramos de atividade
- Auxilia na definição de dados a serem coletados, critérios de seleção e classificação, retenção e distribuição de dados para uso nas organizações
- Utiliza ferramentas de mineração de dados, estatística e probabilística e análise cognitiva e preditiva

APRENDENDO NA PRÁTICA



O seu cliente, agora que entendeu que existem vários perfis profissionais que podem atuar em momentos diferentes do projeto e conheceu as razões desses papéis, está curioso para saber como esses profissionais terão suas atividades distribuídas e controladas.

Você resolveu então, apresentar os conceitos da Engenharia de Software para demonstrar métodos que são aplicados na gestão dos projetos de software e administração da produção do sistema.

Para desenvolver Software

Para **harmonizar trabalhos dos diversos profissionais** envolvidos de forma a garantir a entrega daquilo que é de interesse do usuário final de um programa de computador surge a **Engenharia de Software!**

**INTRODUÇÃO À
ENGENHARIA DE SOFTWARE**

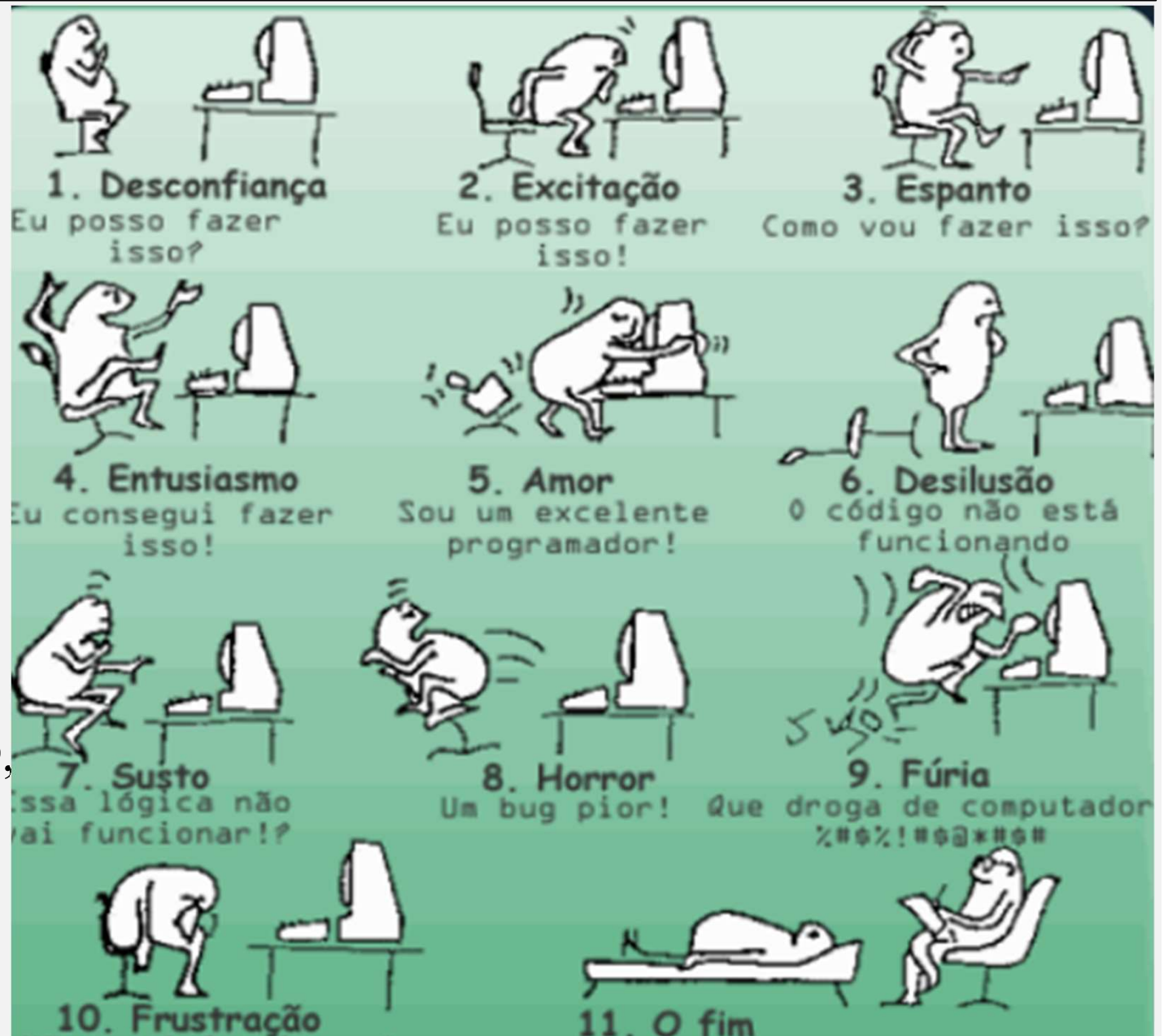
O princípio da Engenharia de Software

A engenharia de software aplica modelos formais precisos que permitam abstrair e documentar requerimentos de software e que permitam ao engenheiro analisar, especificar, projetar, implementar e manter sistemas de software, avaliando e garantindo suas qualidades e possibilitando a capacitação para o uso do software desenvolvido, bem como garantindo as condições técnicas e econômicas que permitam a evolução e ajuste do software.

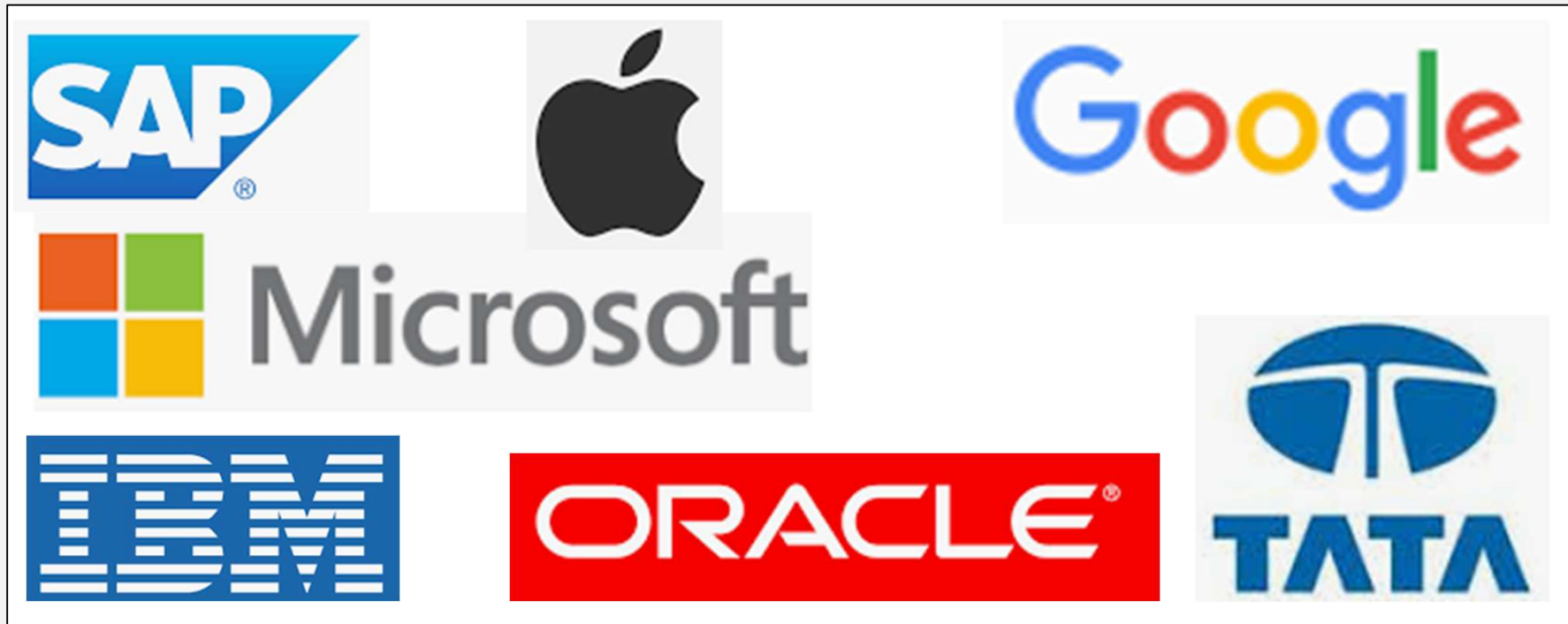
A engenharia de software oferece os métodos para que aconteça o gerenciamento adequado do ciclo de vida de software sob uma estrutura de processos definidos que garantam a formalização documental do software e que apliquem métodos e ferramentas que possibilitem a produção de software de forma sistemática, repetitiva e em escala.

A engenharia de software é portanto “O estabelecimento e uso de sólidos princípios de engenharia para que se possa obter economicamente um software que seja confiável e que funcione eficientemente em máquinas reais.” (Fritz Bauer).

O desafio da Engenharia de Software é **garantir padrões de qualidade no produto final através de um processo formal** que canalize a CRIATIVIDADE humana, criando algo CONCRETO, FUNCIONAL e ÚTIL.



Todas as empresas consagradas que você conhece no ramo de tecnologia utilizam engenharia de software para desenvolver seu produtos!



**Ainda existem exemplos de sistemas sofisticados de automação que operam em missão crítica (não podem parar de funcionar) como os hospitalares, controle de aeronaves, pesquisa espacial, entre outros.*

A adoção de **métodos formais que organizam a produção do software** em suas diversas etapas do processo de manufatura é que **permitem elevar os patamares de qualidade e trazer retorno sobre o investimento!**



A IEEE (Institute of Electrical and Electronic Engineers) publica o SWEBOK (Software Engineering Body of Knowledge) – um guia de práticas de engenharia de software reconhecidas internacionalmente.

Esse material está disponível livremente na Internet.



A VIDA DE UM SOFTWARE

APRENDENDO NA PRÁTICA



Compreendido em linhas gerais que será aplicada a Engenharia de Software, utilizando profissionais de TI de várias especialidades para produzir um sistema que é composto de Hardware, Aplicação, Dados e Processos e Pessoas preparados para operá-lo, o proprietário quer saber se vai valer a pena investir um capital que não deve ser menor que R\$ 100.000,00 no novo sistema.

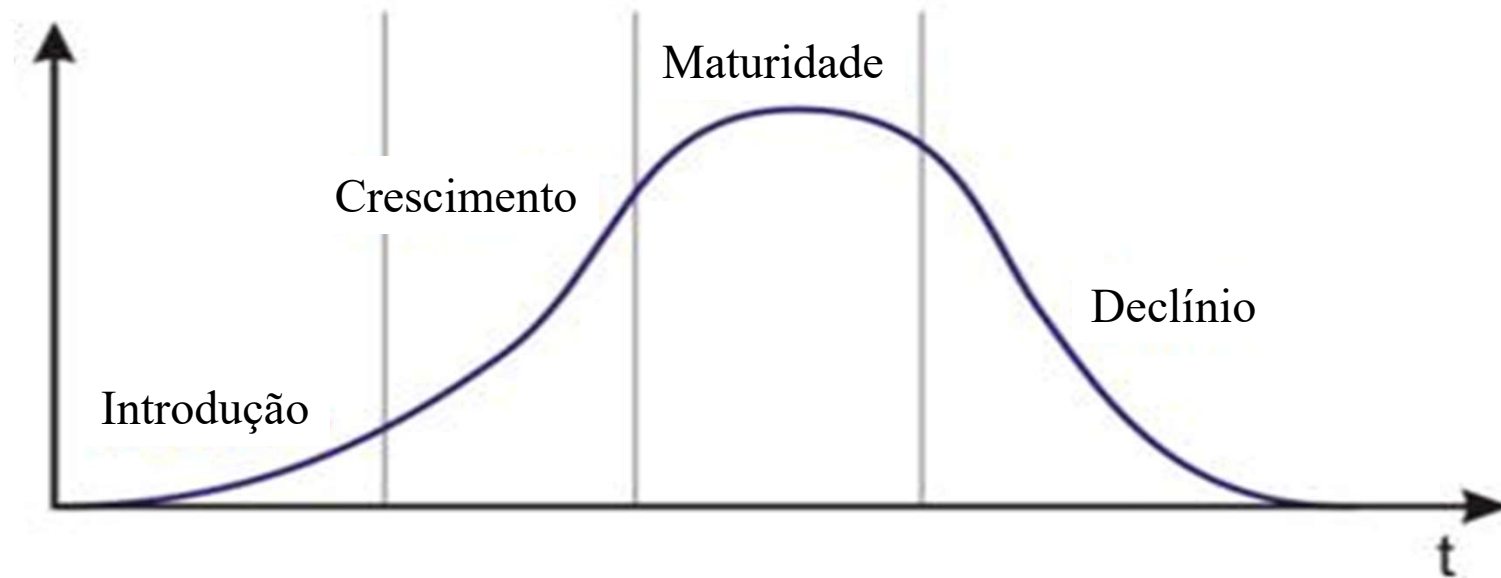
Ele quer saber se uma vez feito esse investimento, nunca mais será necessário investir.

É hora de você explicar para ele como funciona o ciclo de vida de um software...

A base para a Engenharia de Software

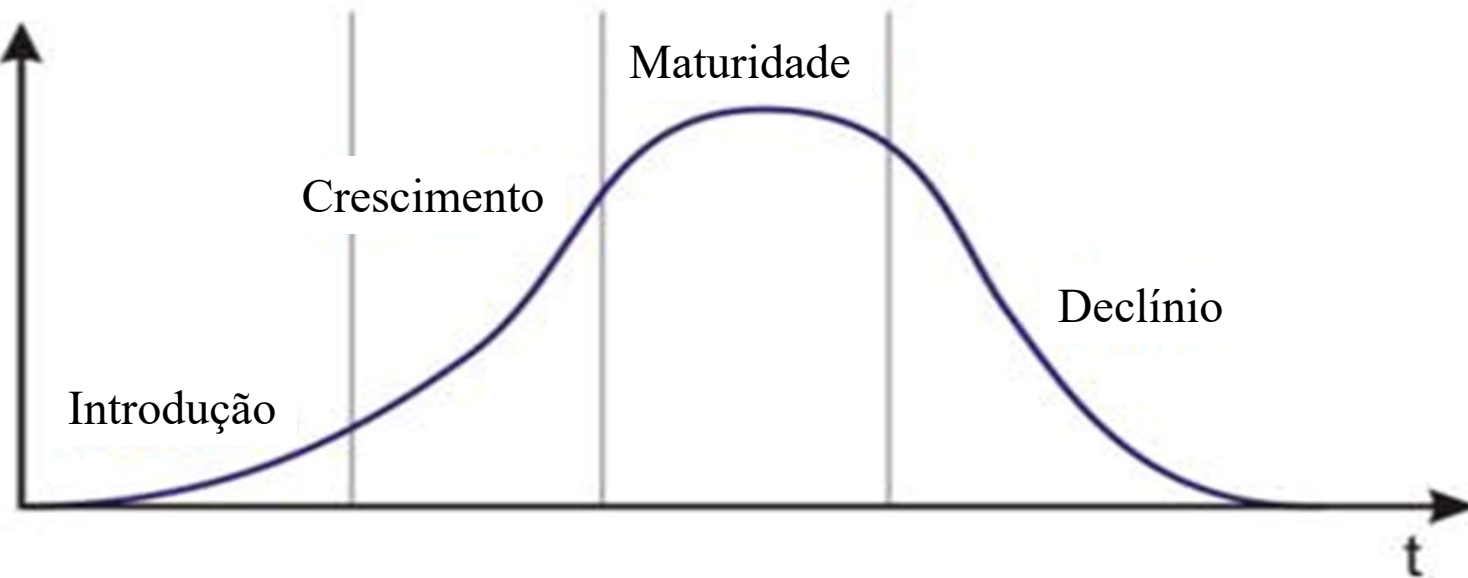
Todo produto tem uma CURVA DE VIDA

Nível de uso e
Remuneração



Essa CURVA DE VIDA é acompanhada por um CICLO DE VIDA que são as Fases pelas quais passa o Software

Nível de uso e
Remuneração



A base para a Engenharia de Software

O Ciclo da Vida de QUALQUER PRODUTO

- Define as **fases pelas quais se passa**, da introdução à aposentadoria
- Define **o que se pode esperar como resultado** do cumprimento de cada fase

Observe a curva de expectativa de vida dos seguintes produtos e responda:

- Produtos de software diferentes têm ciclos de vida diferentes?
- Qual a razão (explique conforme a resposta da questão anterior)?

ESTUDO DE CASO: de quanto em quanto tempo os seguintes produtos recebem uma nova versão (nova geração) com novos recursos funcionais?

- Microsoft Windows
- SAP - ERP

O Ciclo de Vida de Software

De que adianta compreender que o software tem uma vida que passa pela sua produção, entrega, uso e aposentadoria?

Como o gerenciamento do ciclo de vida de software ajuda a INDÚSTRIA DE SOFTWARE?



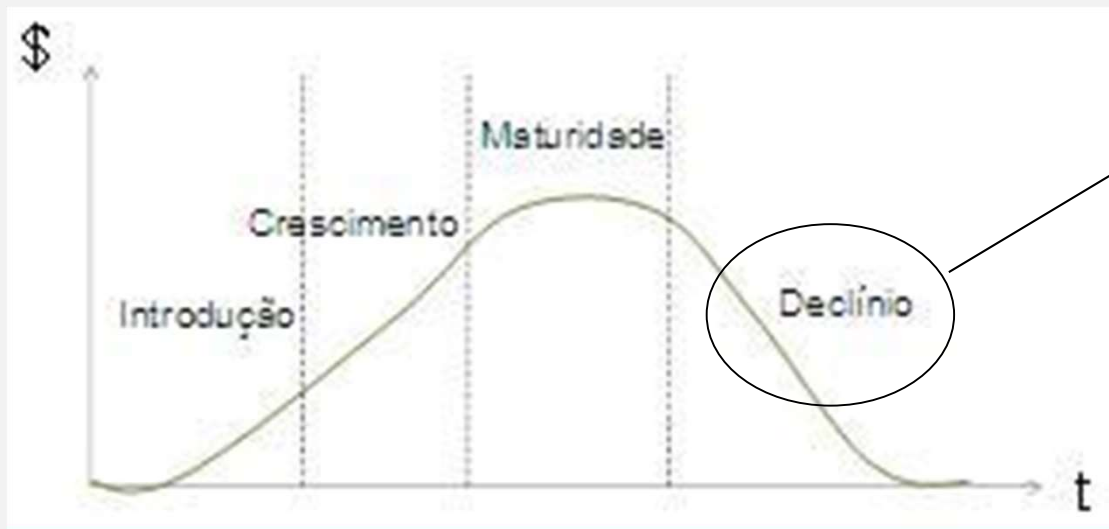
A base para a Engenharia de Software

O Ciclo da Vida bem administrado permite:

- **Retorno do capital investido no desenvolvimento** do produto, gerando lucratividades e sustentabilidade para as fábricas de software;
- **Retorno do capital investido pelo cliente**, que precisa ver melhoria em seus negócios com o uso do software, de forma a pagar o investimento na sua compra e implantação;
- **Manter a fábrica de software** plenamente e permanentemente **ocupada**, otimizando o custo de capital.

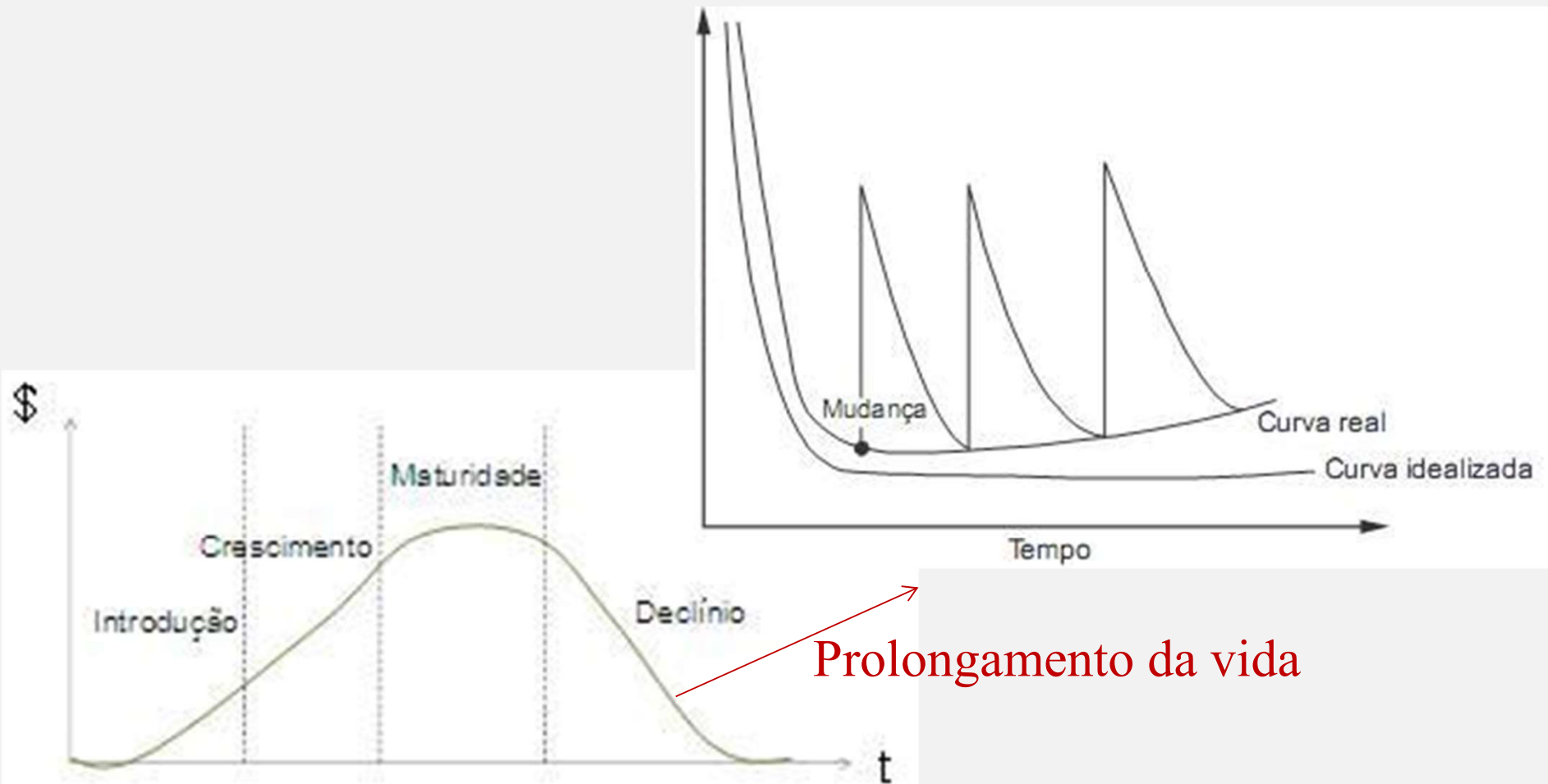
O ciclo de vida pode ajudar no planejamento e obtenção do retorno financeiro esperado.

Seja qual for o modelo de consumo, quanto mais o produto estiver vivo na praça, maior será o retorno que o produto trará!



Como dar sobrevida?

Manutenções e adaptações em produtos prolongam a sua vida.
É o que acontece no versionamento de automóveis e software!



Manutenções podem ser do tipo:

- **Corretivas:** eliminam defeitos do Banco de Dados e programas associados
- **Adaptativas:** ajustam o Banco de Dados e programas associados para acomodar mudanças em regras de negócio que exigem novas funções e informações
- **Evolutivas:** incluem novas estruturas em Bancos de Dados e funcionalidades em programas associados aos Bancos, para criar novas oportunidades de aplicação prática para os usuários, aumentando a atratividade do produto
- **Perfectivas:** buscam ajustes de desempenho (Performance Tuning), otimização de ocupação de espaços de armazenamento de dados e consumo de recursos infraestruturais (processamento, RAM, telecomunicações).

O Ciclo de Vida de Software

Através do estudo de Ciclo de Vida, **toma-se as ações corretas para a preservação da vida** e seu pleno aproveitamento.

No caso de software, o estudo do Ciclo de Vida trata da compreensão e atendimento das necessidades fundamentais de cada etapa do desenvolvimento e uso do software para que aconteça o seu máximo aproveitamento, pelo maior tempo possível, gerando uma relação econômica ideal.

Exemplo do ciclo de vida dos produtos da Microsoft

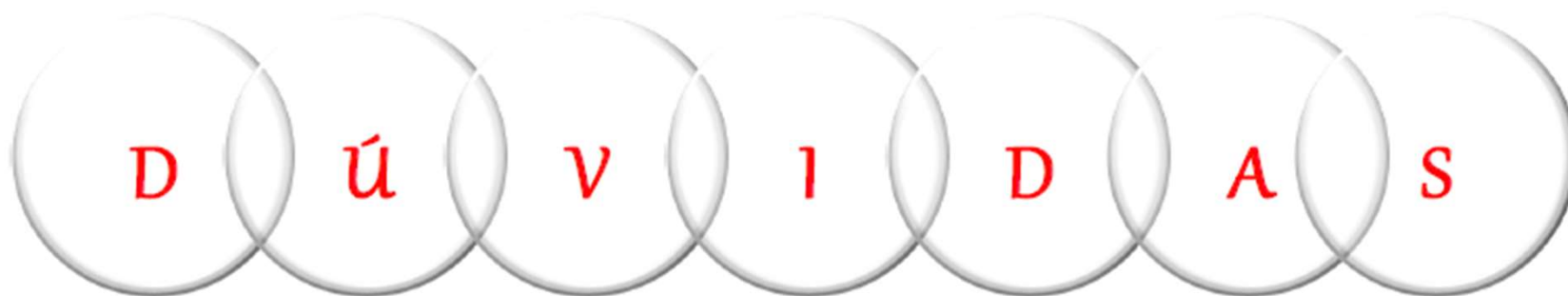
ATIVIDADE EXTRA

Pesquise sobre exemplos de ciclos de vida de softwares (aplicativos, sistemas de informação corporativos, etc.) que você conhece!

Procure por casos que indicam datas de desenvolvimento, lançamento, descontinuação de produtos!

Tome nota e compare os comportamentos dos fabricantes de software!

Faça um julgamento do que você consideraria ideal como tempo de vida e de desenvolvimento para: 1.um sistema de informação que integra todas as atividades de uma empresa; 2.um APP móvel de utilidade pública; 3.um aplicativo de entretenimento ou jogo.



Material de aula estará no site após a aula.

BONS ESTUDOS!

Bibliografia

- PRESSMAN, R. S. Engenharia de software. São Paulo: Editora McGraw-Hill, 2002.
- HIRAMA, KECHI. Engenharia de Software: qualidade e produtividade com tecnologia. Editora Elsevier, Rio de Janeiro, 2011.
- SOMMERVILLE, IAN. Engenharia de software. Editora Pearson, 9.ed. - São Paulo, 2014.
- PFLEEGER, S. LAWRENCE. Engenharia de Software: teoria e prática. Editora Pearson, 2. ed. - São Paulo, 2004. FILHO, MILTON M. A. Fundamentos do Desenho Orientado a Objetos com UML. Editora Makron Books, São Paulo, 2001.