



**INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**
Rio de Janeiro

Disciplina

Modelagem de Domínio e
Conceitos de Orientação a Objetos

PROF. FERNANDO TAMBERLINI ALVES

- Apresentação da disciplina
- Ementa e Objetivo da Disciplina
- O que é Modelo, Domínio e Orientação a Objetos?
- Atividade Prática
- Referências

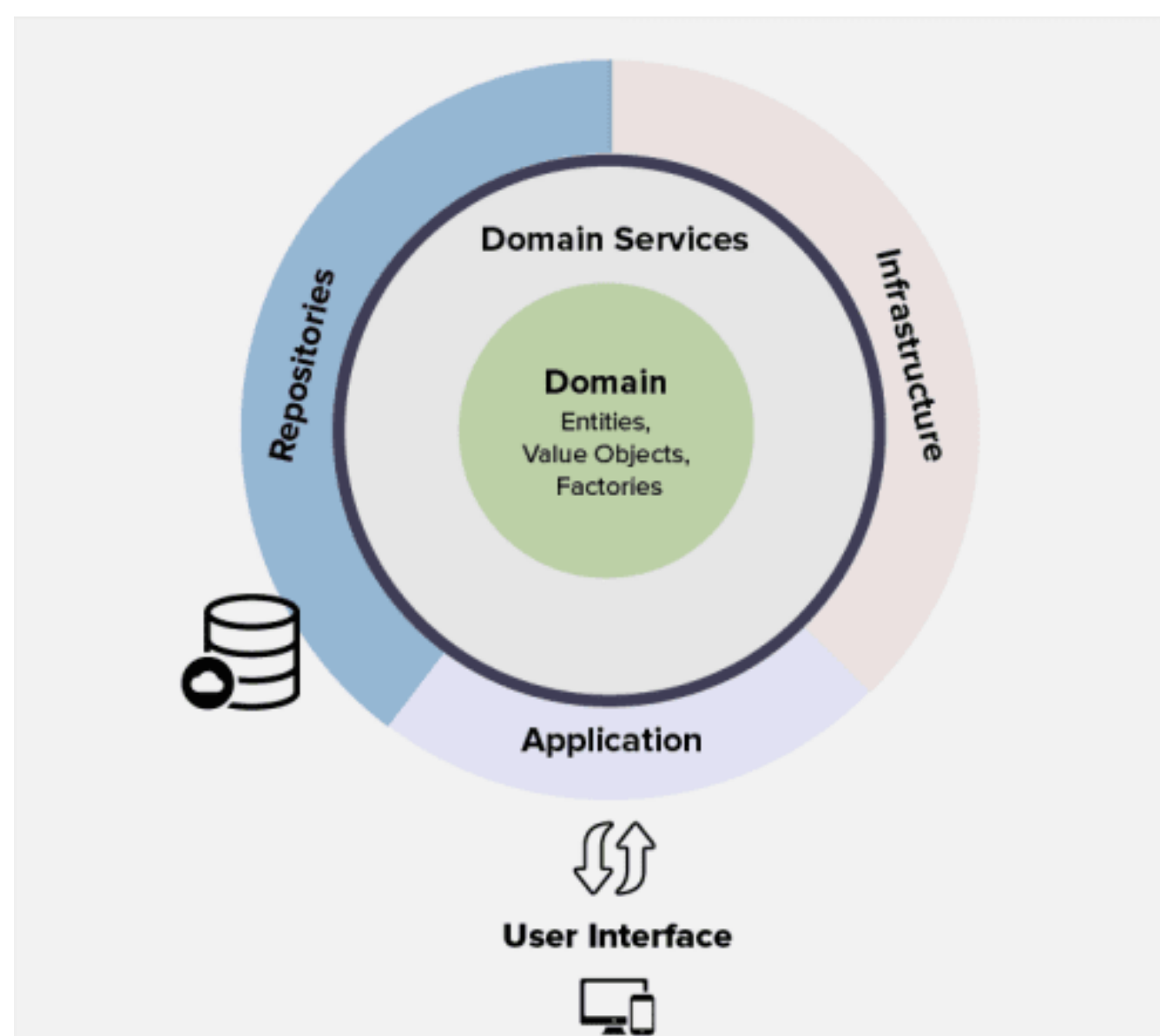
Disciplina MDCCOO – 2024.01

- Modelagem de Domínio e Conceitos de Orientação a Objetos
- Prof. Fernando Tamberlini Alves
E-mail: fernando.alves@ifrj.edu.br
Site: ftamberlini.dev.br
- Horário: 2ª feira – 08:30 até 11:45 – Matutino
2ª feira – 13:15 até 16:15 – Vespertino
- Local: Laboratório 203 – IFRJ/CSJM

Ementa e Objetivo Geral

Modelagem de domínio e o paradigma orientado a objetos.

Desenvolver competências de modelagem de domínio e reconhecer conceitos do paradigma orientado a objetos



Programação Orientada a Objetos

Polimorfismo

Herança

Abstração

Encapsulamento



Objetivos Específicos

- Reconhecer os conceitos gerais de “Domínio”, “Abstração”, “Modelo” e “Representação”
- Identificar construtores básicos de Diagramas de Classe UML: Classes, Atributos e Relacionamentos
- Criar modelos de classe utilizando Diagramas de Classe UML
- Reconhecer os conceitos específicos sobre o paradigma orientado a objetos: “Classe”, “Objeto”, “Herança”, “Composição”, “Polimorfismo” e “Encapsulamento”
- Identificar a sintaxe, semântica e instruções básicas de uma linguagem de programação orientada a objetos
- Implementar modelos de domínio com uma linguagem de programação orientada a objetos

Planejamento do Curso

1º Bimestre

Unidade I – Modelagem de Domínio

- Conceitualização de “Domínio”, “Abstração”, “Modelo” e “Representação”
- Construtores básicos de Diagramas de Classe UML
- Criação de modelos de domínio representados com Diagramas de Classe UML

2º Bimestre

Unidade II – O Paradigma Orientado a Objetos

- Conceitualização de Classe”, Objeto”, “Herança”, “Composição”, “Polimorfismo” e “Encapsulamento”
- Aplicação dos conceitos em Diagramas de Classe UML
- Sintaxe, semântica e instruções básicas de uma linguagem de programação orientada a objetos
- Implementação de modelos de domínio utilizando uma linguagem de programação orientada a objetos

Metodologia e Recursos

- Desenvolvimento Metodológico:

Aula expositiva e prática, a partir do uso de aplicativos e ferramentas da internet, com foco na colaboração. Atividades práticas envolvendo estudos de caso, seminários, trabalhos individuais e em grupo, apresentação individual e em grupo.

- Recursos:

Material digital, datashow, laboratório de informática com acesso à internet para o professor e para os alunos, preferencialmente um computador por aluno.

- Ferramentas:

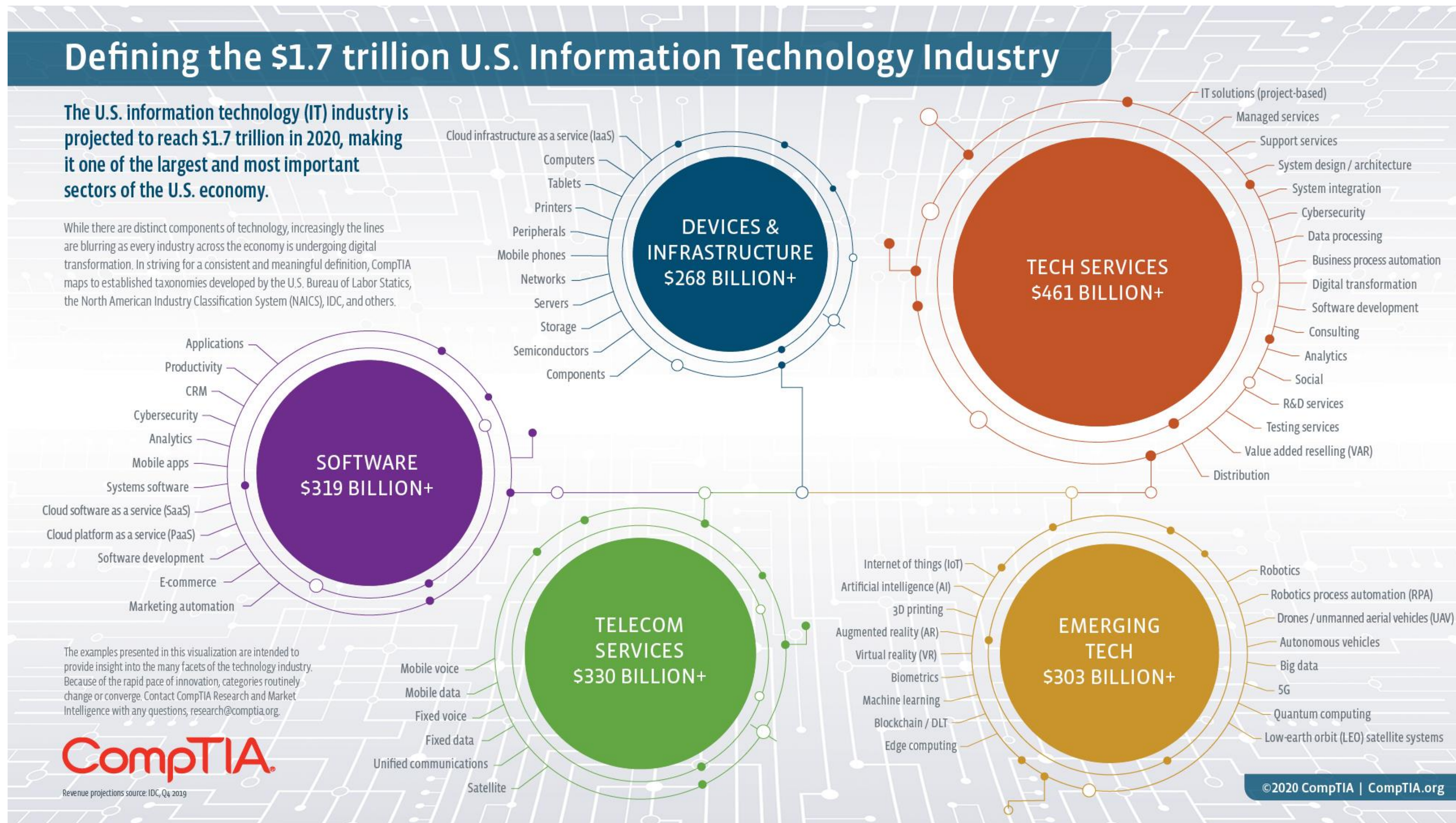
Sistema Operacional Windows, VSCode, Notepad++, LucidChart e Draw.io e Ferramentas de Inteligência Artificial.

- 1º Bimestre:
 - 1ª Avaliação – Prova sobre Modelagem de Domínio e Diagrama UML
 - 2ª Avaliação – Trabalho Dupla – Elaboração de Diagrama de Classes
 - $MV1 = (AV1 + AV2 + AV3)/3$
- 2º Bimestre:
 - 4ª Avaliação – Trabalho Dupla – Elaboração de Diagrama de Classes
 - 5ª Avaliação – Trabalho Dupla – Codificação de Jogo com POO
 - 6ª Avaliação – Prova Individual – Orientação a objetos
 - $MV2 = (AV4 + AV5 + AV6)/3$
- Grau Final
 - $G = (MV1 + 2 * MV2) / 3$
 - Maior ou igual a 6,0 – Aprovado
 - Menor que 6,0 – Recuperação

Bibliografia Básica

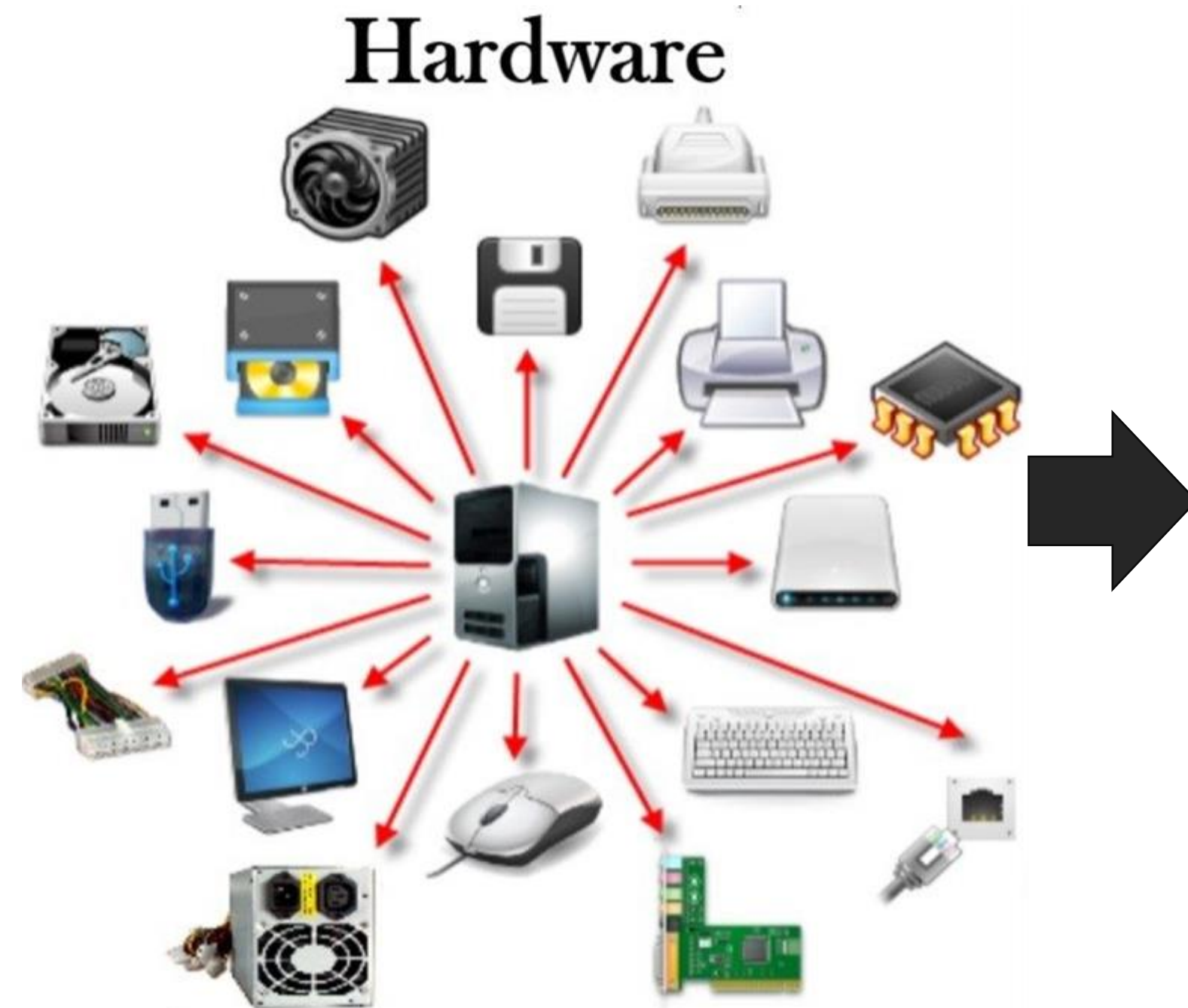
- ANICHE, M. (2015). Orientação a Objetos e SOLID para Ninjas: Projetando classes flexíveis. Editora Casa do Código
- GUEDES, G. T. (2018). UML 2-Uma abordagem prática. Novatec Editora.
- LEITE, T. (2016). Orientação a objetos: aprenda seus conceitos e suas aplicabilidades de forma efetiva. Editora Casa do Código.

Setores de TI



Setores de TI x Disciplinas

Infraestrutura



Desenvolvimento



Software consiste em:

- (1) instruções (programas de computador) que, quando executadas, fornecem características, funções e desempenho desejados;
- (2) estruturas de dados que possibilitam aos programas manipular informações adequadamente e;
- (3) informação descritiva, tanto na forma impressa quanto na virtual, descrevendo a operação e o uso dos programas.

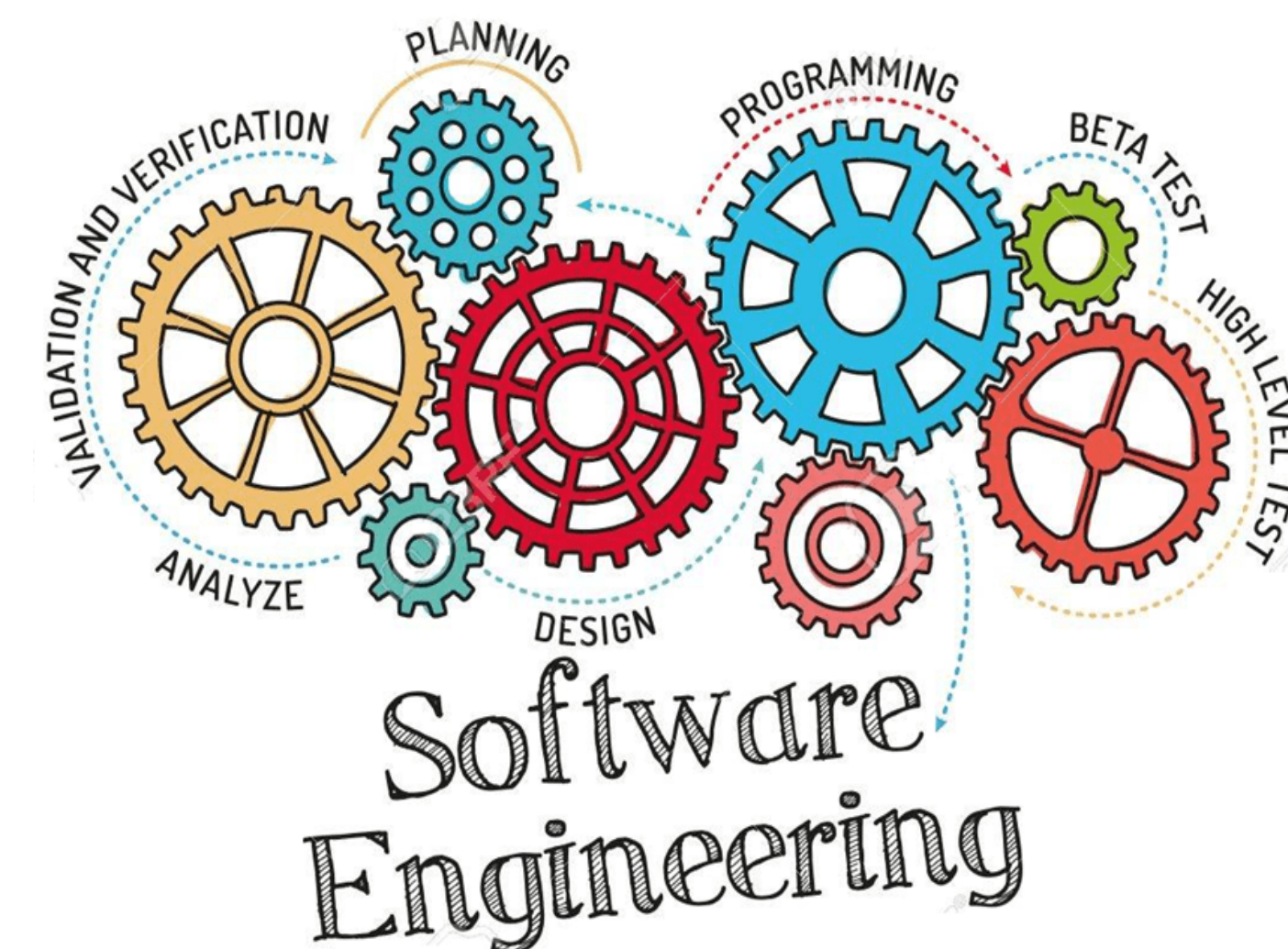
Ref: Engenharia de software - Roger S. Pressman; Bruce R. Maxim

Da Análise à Eng. Software

A engenharia de software engloba processos, métodos e ferramentas que possibilitam a construção de sistemas complexos baseados em computador dentro do prazo e com qualidade.

O processo de software incorpora cinco atividades estruturais: comunicação, planejamento, modelagem, construção e entrega, e elas se aplicam a todos os projetos de software.

A prática da engenharia de software é uma atividade de resolução de problemas que segue um conjunto de princípios básicos.



Ref: Engenharia de software - Roger S. Pressman; Bruce R. Maxim

Desafios do Processo de Software

Facemash

Facemash



Mark Zuckerberg created Facebook in his Harvard dorm room.

Facemash, the Facebook's predecessor, opened on October 28, 2003^[1]. Initially, the website was invented by a Harvard student, Mark Zuckerberg, and three of his classmates – Andrew McCollum, Chris Hughes and Dustin Moskovitz. Zuckerberg wrote the software for the Facemash website when he was in his second year of college. The website was set up as a type of "hot or not" game for Harvard students. The website allowed visitors to compare two student pictures side-by-side and let them choose who was "hot" and who was "not".....

That night, Mark Zuckerberg wrote the following blog entries:^{[2][3][4]}

I'm a little intoxicated, not gonna lie. So what if it's not even 3 pm and it's a Tuesday afternoon? What? The Kirkland dormitory facebook is open on my desktop and some of these people have pretty horrendious facebook pics. I almost want to put some of these faces next to pictures of some farm animals and have people vote on which is more attractive.

—2:49 pm

Yea, it's on. I'm not exactly sure how the farm animals are going to fit into this whole thing (you can't really ever be sure with farm animals...), but I like the idea of comparing two people together.

—11:10 am

Let the hacking begin.

—2:57 pm

By Colsen: I posted awhile back. Support Mark Zuckerberg!

—11:00 pm

FACEMASH

n for our looks? No. Will we be judged on them? Yes.

Who's Hotter? Click to Choose.



Amy

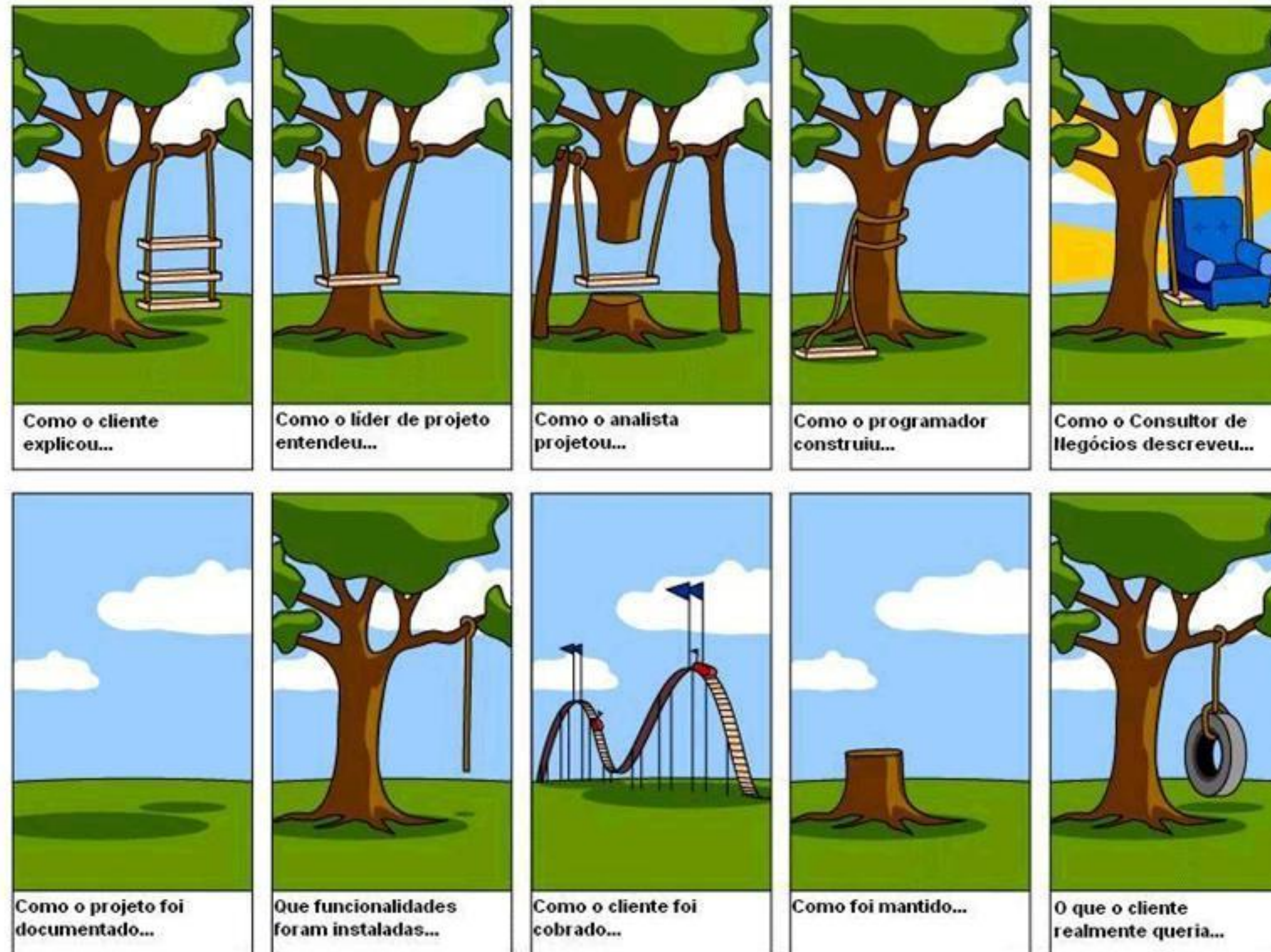


Maria

Desafio de Instruções Exatas



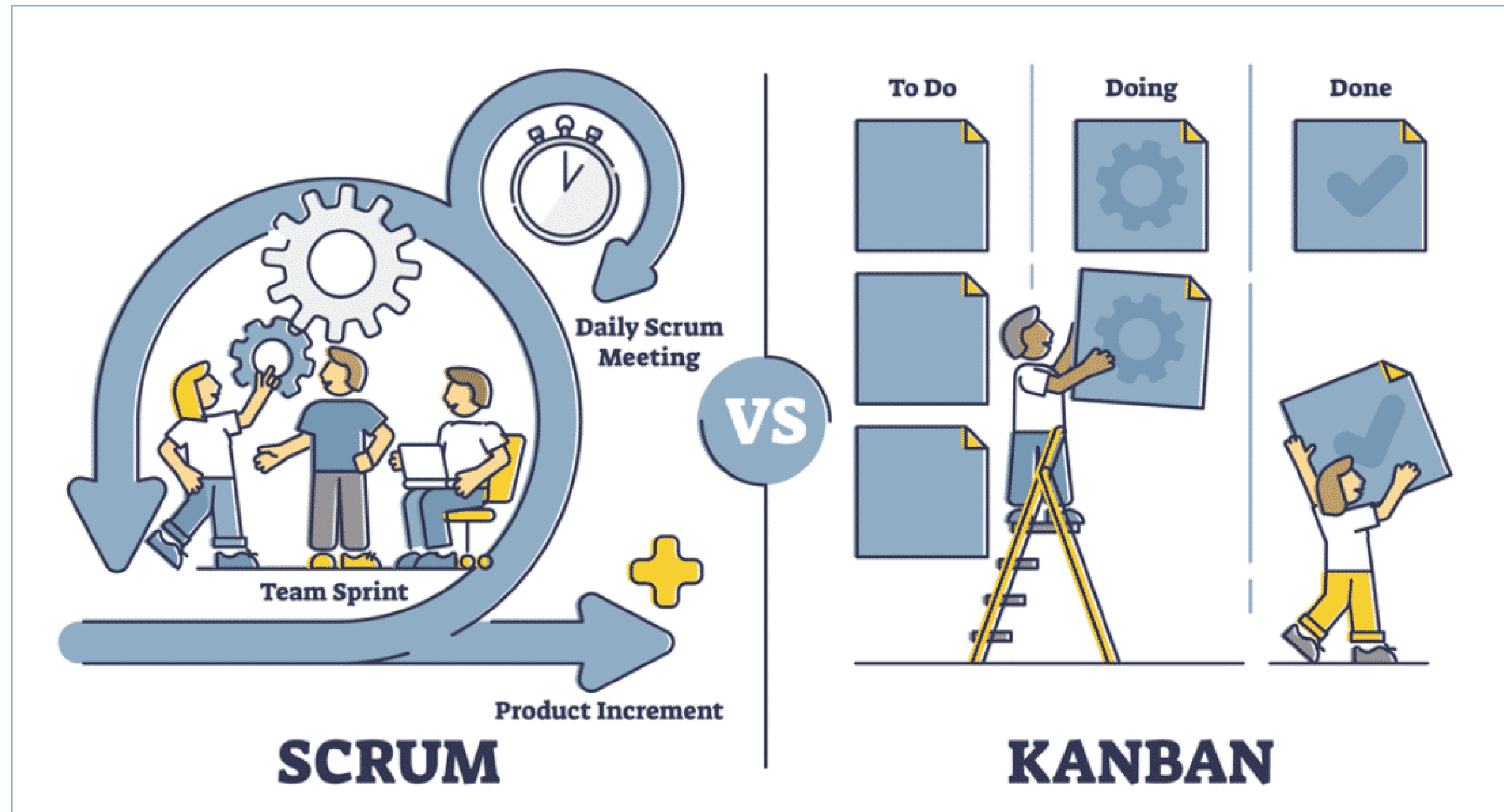
Piadinha Clássica...



MANIFESTO ÁGIL



Metodologia Ágil



Metodologia Ágil

Iterativo



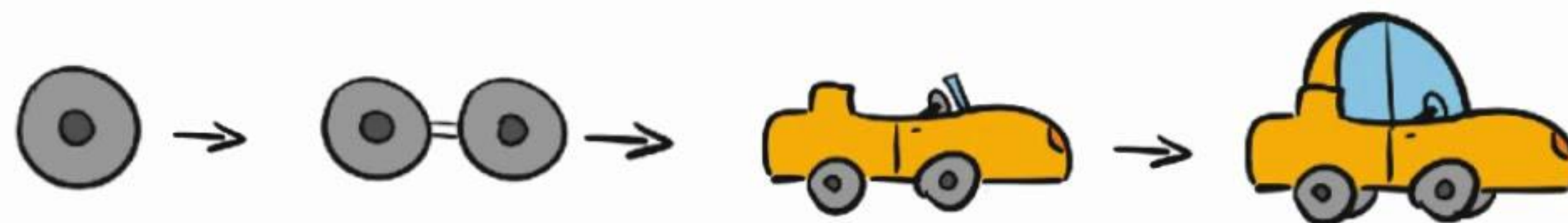
Incremental



Iterativo e Incremental

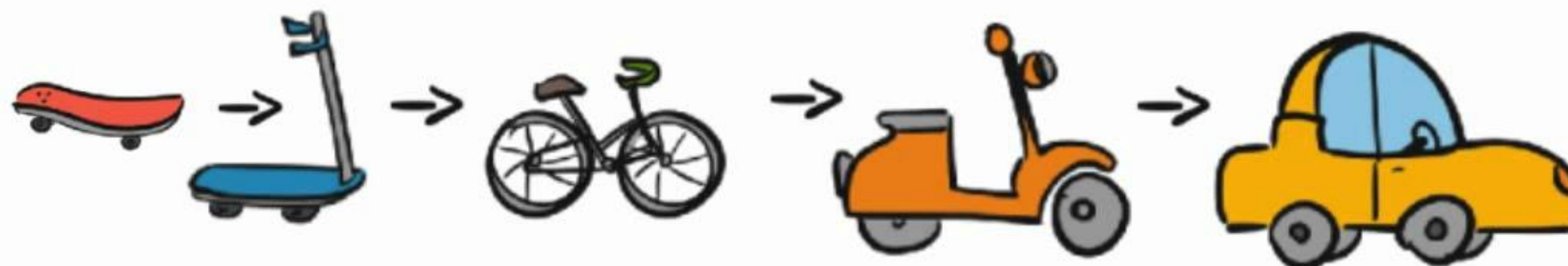


Metodologia Ágil



MÓDELO ITERATIVO

MÓDELO ITERATIVO E INCREMENTAL



Breve explicação sobre HTML...

HTML



HTML – HyperText Markup Language

- Traduzindo para “Linguagem de Marcação de Hipertexto”
- O que é Hipertexto?
- O que é Marcação (Tags)?
- É uma linguagem de programação?
- Quem lê um arquivo HTML?
- Como ver o HTML de uma página na WEB?

1. Abrir o editor de texto Notepad++ e selecionar a linguagem HTML no menu superior “Linguagem”;
2. Digitar a conteúdo do arquivo “aula01.html” exposto no projetor;
3. Salvar o arquivo “aula01.html” na área de trabalho e abri-lo em algum navegador;
4. Explorar o formulário do arquivo criado e apontar os seus erros.



A análise de domínio está relacionada à descoberta das informações gerenciadas pelo sistema a ser construído, ou seja, à representação e a transformação da informação inerente ao processo objeto do sistema.

Ref: Análise e Projeto de Sistemas de Informação Orientados a Objetos
Raul Sidnei Wazlawick

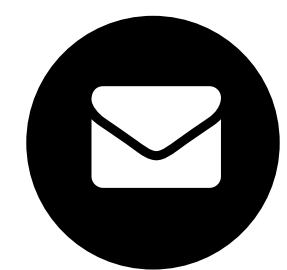
Os modelos de requisitos (também denominados modelos de análise) representam os requisitos dos clientes, descrevendo o software em três domínios diferentes: o domínio da informação, o domínio funcional e o domínio comportamental.

Ref: Engenharia de software - Roger S. Pressman; Bruce R. Maxim

Dúvidas?



Dúvidas?



fernando.alves@ifrj.edu.br



ftamberlini.eti.br/ifrj