Programação Orientada a Objetos

Prof. Dr. Josenalde Barbosa de Oliveira

josenalde.oliveira@ufrn.br

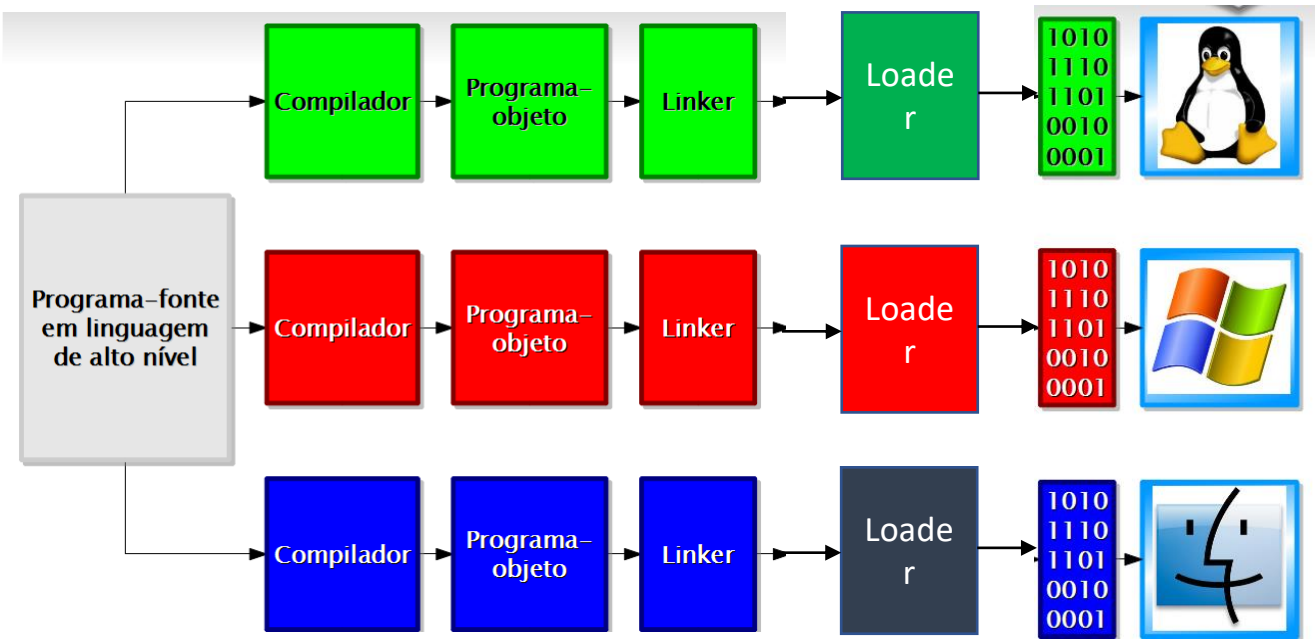
https://github.com/josenalde/apds

Linguagens interpretadas x compiladas

Compilação *(Exemplos C, C++): conjunto de etapas: pré-processamento (código fonte é expandido, pela abertura das diretivas #), verificação sintática, para cada arquivo fonte (exemplo .c) é criado um arquivo objeto, com instruções de linguagem de máquina que correspondem ao arquivo fonte compilado; por último, a link-edição une todos os arquivos objeto num único arquivo executável (fonte + objetos das bibliotecas usadas) – compilação estática*

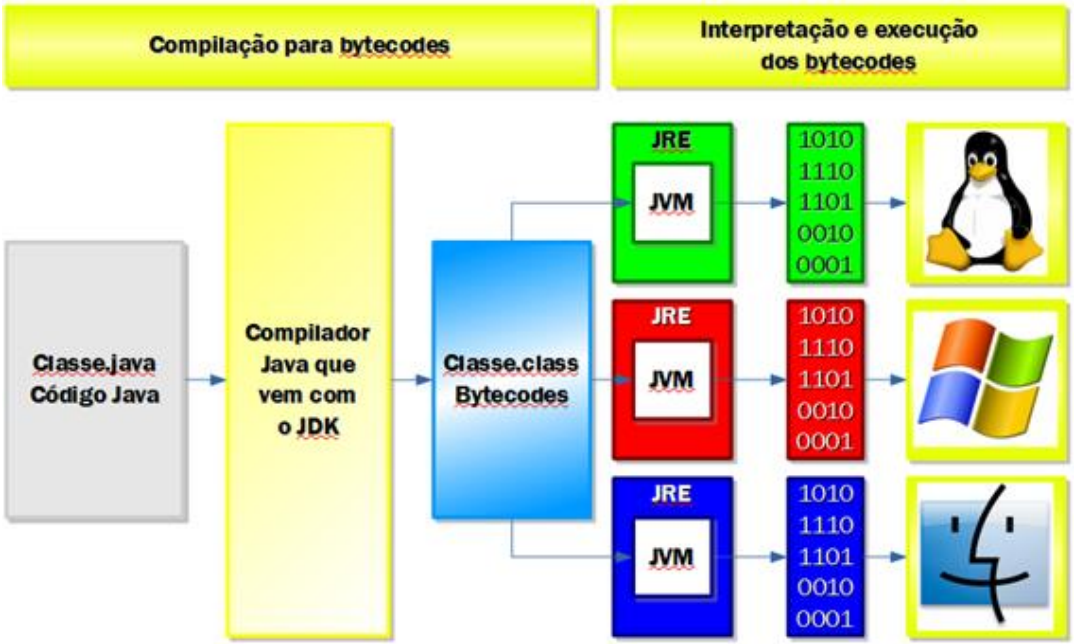
Interpretação*: (Exemplos Javascript, Python, Matlab): tem seus códigos fonte transformados em linguagem intermediária que será interpretada pela máquina virtual da linguagem quando for executada. Este processo consiste na tradução da ling. intermediária para a linguagem da máquina virtual – "compilação dinâmica"*

Linguagens interpretadas x compiladas



https://rogeraoaraujo.com.br/2013/01/06/java-compilacao-de-classes-java-parte-i/

Linguagens interpretadas x compiladas



JDK – compilador: .java->.class

JRE – ambiente de execução na plataforma alvo (JVM+API Java)

JVM – interpreta e executa o bytecode

A **máquina virtual Java** é a responsável pela **portabilidade**! Ela interpreta e executar o bytecode. É a provedora de formas e meios de o aplicativo conversar com o sistema operacional.

API (Java Application Programming Interface):É uma biblioteca de componentes que possui vários recursos úteis; e é utilizada para execução de aplicações Java.

https://rogeraoaraujo.com.br/2013/01/21/java-compilacao-de-classes-java-parte-ii/

Linguagens interpretadas x compiladas

FGV 2012 Senado Federal – Prova anulada – Análise de Sistemas – Questão 55] Para permitir que um mesmo programa seja executado em vários sistemas operacionais, a plataforma java gera códigos genéricos *.class e os traduz para o código da máquina local, *.exe ou *.bin, somente no momento da execução. Nesse contexto, os códigos específicos para a máquina virtual Java, e não para a máquina local, recebe o nome de:

- [A] microcode.
- [B] scriptcode.
- [C] framecode.
- [D] bytecode.
- [E] javacode.

https://rogeraoaraujo.com.br/2013/01/21/java-compilacao-de-classes-java-parte-ii/

Linguagens interpretadas x compiladas

[FCC 2010 TRT 9ª Região – Técnico Judiciário – Especialidade Tecnologia da Informação – Questão 36] O JVM mais o núcleo de classes da plataforma Java e os arquivos de suporte formam o

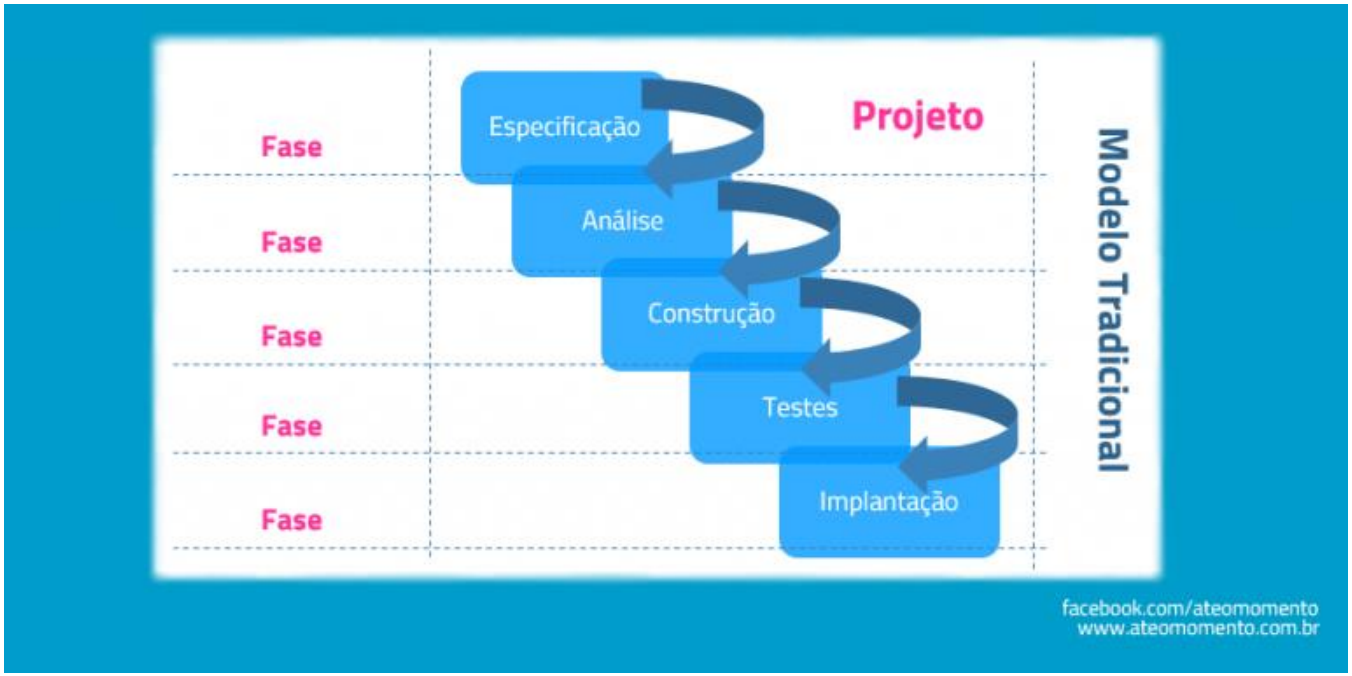
- [A] o J2EE.
- [B] o JDK.
- [C] o JRE.
- [D] uma JSP.
- [E] uma API.

FCC 2007 TJ/PE – Analista Judiciário – Analista de Suporte – Questão 25] O código Java compilado é gerado em arquivo com extensão

- [A] .ser
- [B] .jar
- [C] .java
- [D] .html
- [E] .class

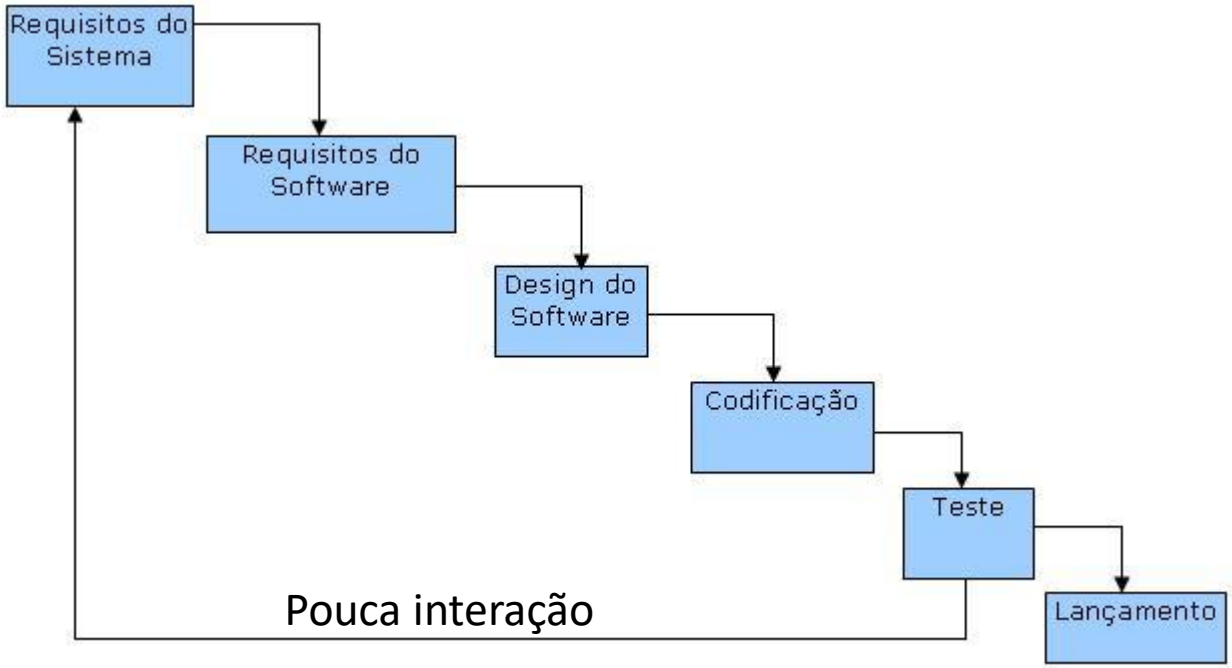
https://rogeraoaraujo.com.br/2013/01/21/java-compilacao-de-classes-java-parte-ii/

2. Metodologias de desenvolvimento



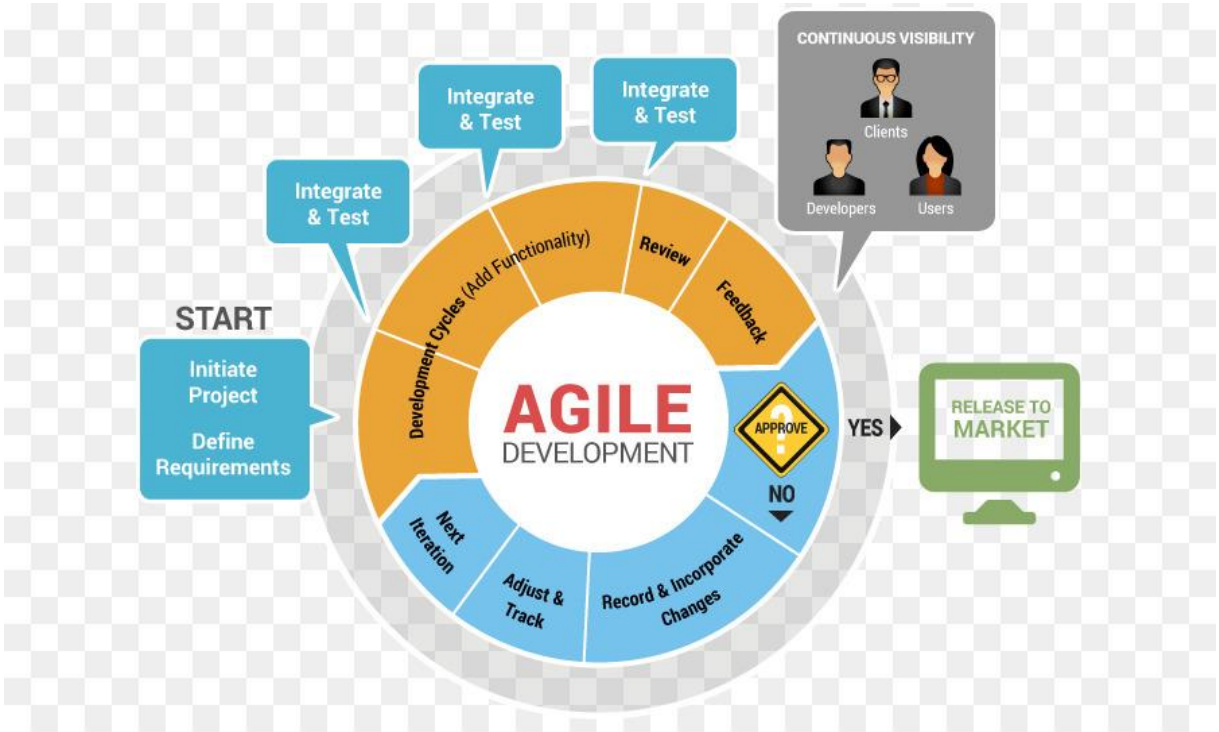
Modelo cascata – fases sequenciais, com envolvimento do cliente nas fases de especificação/ planejamento e análise, e só mais adiante na fase de implantação; a metodologia mais usada atualmente **são os métodos ágeis**, com foco no cliente e entregas funcionais mais rápidas ao Longo de todo o processo! (ver extreme programming (xp) e agile programming)

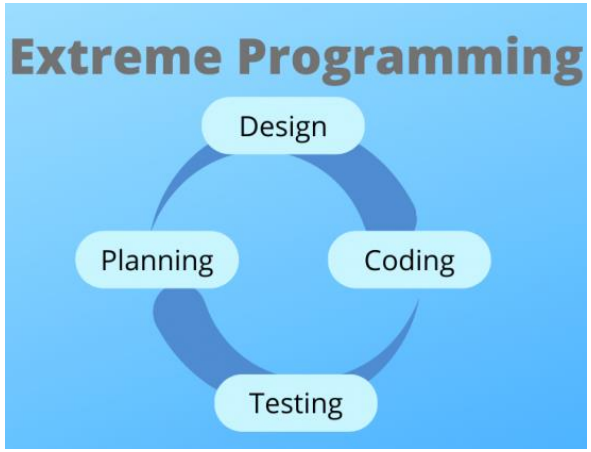
2. Metodologias de desenvolvimento



Modelo cascata – fases sequenciais, com envolvimento do cliente nas fases de especificação/ planejamento e análise, e só mais adiante na fase de implantação; a metodologia mais usada atualmente **são os métodos ágeis**, com foco no cliente e entregas funcionais mais rápidas ao Longo de todo o processo! (ver extreme programming (xp) e agile programming)

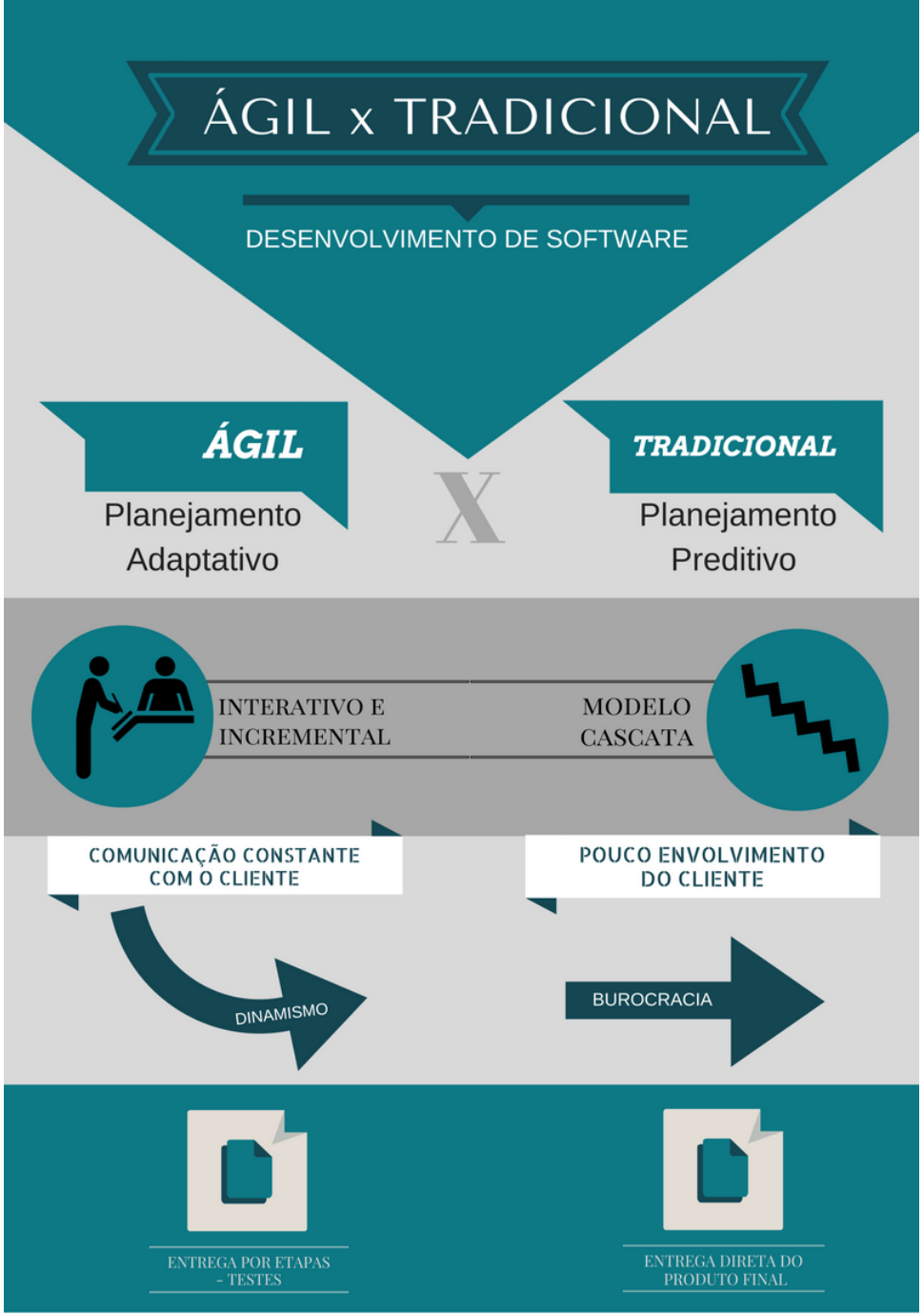
2. Metodologias de desenvolvimento





2. Metodologias de desenvolvimento

- SCRUM (pesquisar)
- MVP (pesquisar)
- TRL (pesquisar)
- PMBoK



ateliê de software

2. Metodologias de desenvolvimento

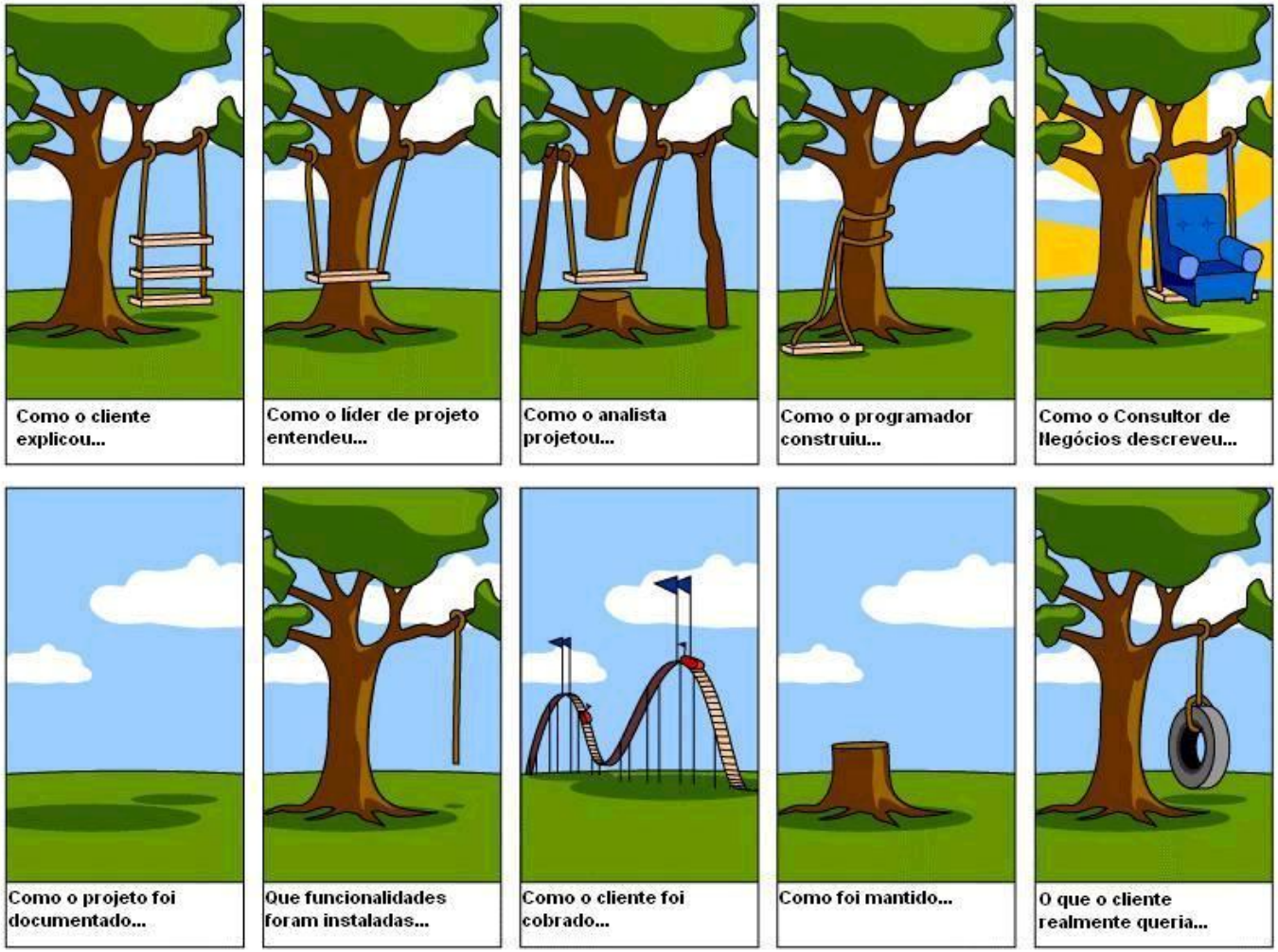
- Para definição e acompanhamento do processo de desenvolvimento do projeto utiliza-se ferramentas baseadas no SCRUM/KANBAN (lista de tarefas; fazer, fazendo, feito)

Trello

3. Levantamento de requisitos

- No domínio que estivermos tentando modelar (por conversas com especialistas, experiências anteriores, livros, manuais, pesquisas, questionários etc.), é preciso identificar os requisitos.

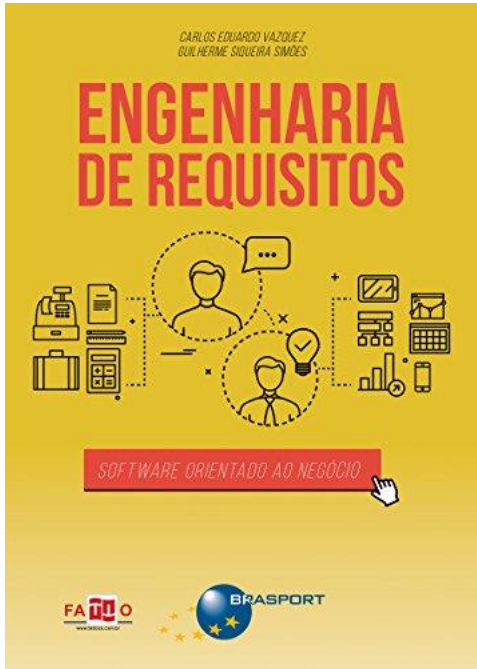
- Um **REQUISITO** é aquilo que é de suma importância constar como tarefa a ser desenvolvida no projeto.



3. Levantamento de requisitos

- OBJETIVOS
  - Identificar as funções que o sistema deverá realizar (FUNCIONAIS, RF), independem da implementação ou da tecnologia utilizada.
  - Identificar requisitos que se referem à confiabilidade do sistema, ao tempo de resposta, à tolerância a falhas etc. Descreve por exemplo quais parâmetros computacionais devem ser atendidos pelo Sistema (NÃO FUNCIONAIS, RNF)

- Exemplos:
  - RF01: o sistema deve permitir a inclusão, alteração, busca e inclusão dos clientes de uma empresa (CRUD)
  - RF02: o sistema deve permitir gerar a folha de pagamento
  - RNF01: o banco de dados do sistema deve possuir níveis de segurança de acesso
  - RNF02: os módulos de terceiros utilizados no sistema devem ser de código aberto
  - RNF03: o tempo de autorização do cartão de crédito não pode ultrapassar 5s



3. Levantamento de requisitos

- REGRAS DE NEGÓCIO
  - As regras de negócio (RN) são restrições importantes que o domínio impõe e que precisam ser respeitadas para que o requisite functional seja cumprido.

- Exemplos:
  - RN01: um cliente deve possuir um número de CPF válido
  - RN02: um funcionário deve possuir um salário cadastrado maior ou igual ao salário mínimo vigente
  - RN03: O código, nome e valor de um produto não podem estar em branco
  - RN04: o nome e o código de um produto não podem ser repetir
  - RN05: deve haver um cliente associado à venda
  - RN06: um funcionário é o responsável pela venda

- ARTEFATO (RESULTADO)
  - **DOCUMENTO DE REQUISITOS**
  - **Exemplo:** **https://docs.google.com/document/d/169gqhewVyVmXnj2AMLRP5IdnjnlcjUyxk4aH_TY0wcY/edit**

3. Levantamento de requisitos

Fonte: https://analisederequisitos.com.br/requisitos-funcionais-e-nao-funcionais/

3. Levantamento de requisitos

- Mais Exemplos de RNF (é preciso ser mensurável):
  - O sistema deve ser multiplataforma – Windows, Linux e macOS.
  - O desenvolvimento deve ser em linguagem Java
  - O programa deve funcionar offline.
  - O sistema deve respeitar o tempo máximo de 160 segundos durante processamentos

- Estes exemplos não são RNF:
  - O sistema deve ser rápido
  - Não deve corromper dados
  - O sistema deve ser seguro

3. Levantamento de requisitos

3. Levantamento de requisitos

Exercício: elaborar um documento de requisitos para um sistema de gerenciamento de biblioteca de uma entidade de ensino

Visão geral do sistema

O sistema deve gerenciar os processos de uma biblioteca. Permitindo o cadastro de itens e usuários. Empréstimo e devolução de itens, bem como, emissão de relatórios.