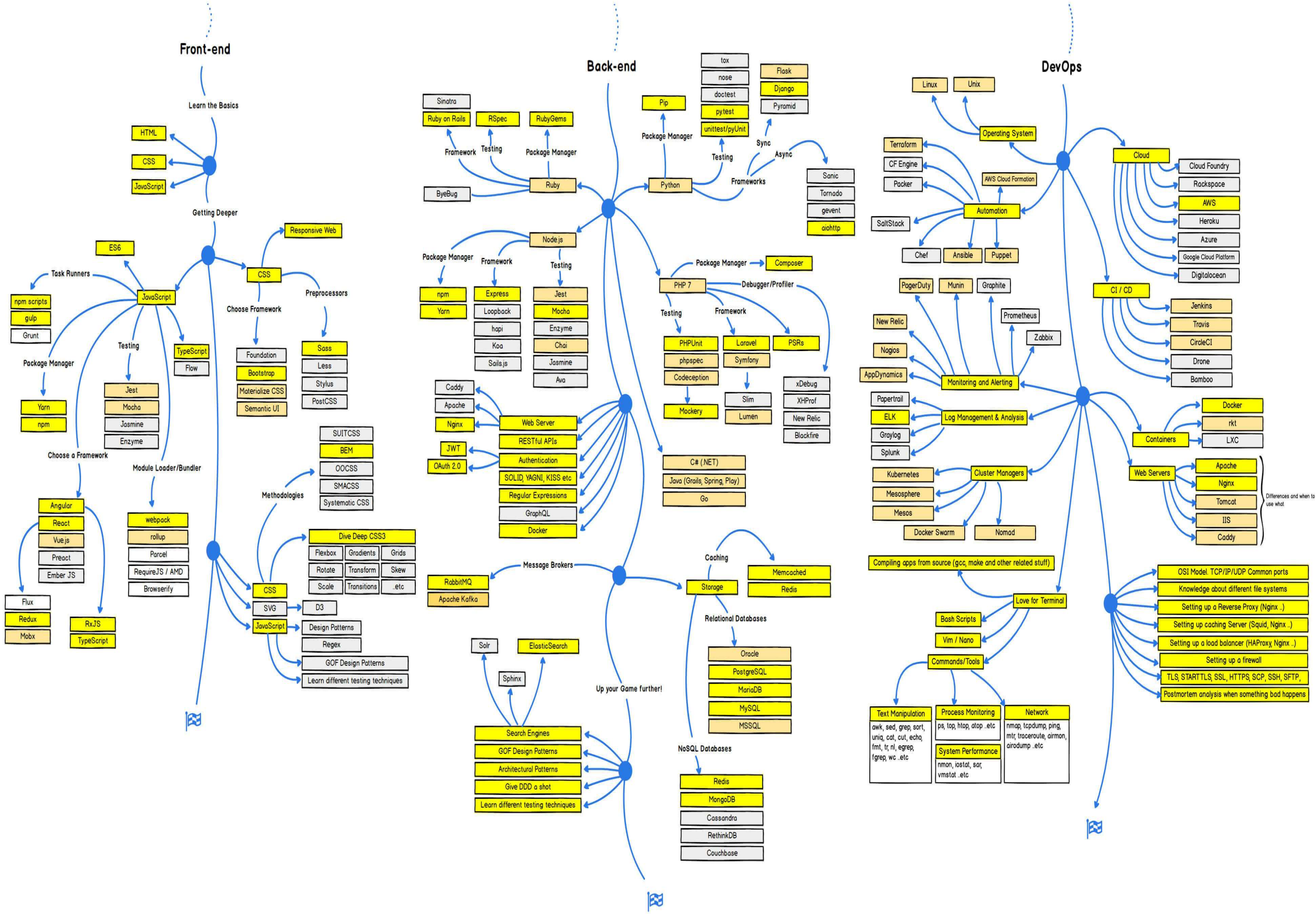
INTEGRANDO REQUISITOS,
DESENVOLVIMENTO E TESTES EM JAVA

BDD

ACHILES LUCIANO

- Analista de Testes “novato”
- Mestre em Ciência da Computação UFCG
- Areiense -PB
- Motociclista nas horas vagas


Front-end
Learn the Basics
HTML
CSS
JavaScript
Getting Deeper
ES6
Task Runners
npm scripts
gulp
Grunt
JavaScript
Package Manager
Yarn
npm
Testing
Jest
Mocha
Jasmine
Enzyme
TypeScript
Flow
Choose a Framework
Angular
React
Vue.js
Preact
Ember JS
Flux
Redux
Mobx
RxJS
TypeScript
Module Loader/Bundler
webpack
rollup
Parcel
RequireJS / AMD
Browserify
CSS
Responsive Web
Choose Framework
Foundation
Bootstrap
Materialize CSS
Semantic UI
Preprocessors
Sass
Less
Stylus
PostCSS
Methodologies
SUITCSS
BEM
OOCSS
SMACSS
Systematic CSS
CSS
Dive Deep CSS3
Flexbox
Gradients
Grids
Rotate
Transform
Skew
Scale
Transitions
..etc
SVG
D3
JavaScript
Design Patterns
Regex
GOF Design Patterns
Learn different testing techniques
Back-end
Sinatra
Ruby on Rails
RSpec
RubyGems
Framework
Testing
Package Manager
ByeBug
Ruby
Pip
Package Manager
tox
nose
doctest
pytest
unittest/pyUnit
Testing
Flask
Django
Pyramid
Sync
Async
Frameworks
Python
Sanic
Tornado
gevent
aiohttp
Node.js
Package Manager
Framework
Testing
npm
Yarn
Express
Loopback
hapi
Koa
Sails.js
Jest
Mocha
Enzyme
Chai
Jasmine
Ava
Package Manager
Composer
PHP 7
Debugger/Profiler
Testing
Framework
PHPUnit
phpspec
Codeception
Mockery
Laravel
Symfony
Slim
Lumen
PSRs
xDebug
XHProf
New Relic
Blackfire
Caddy
Apache
Nginx
Web Server
JWT
OAuth 2.0
RESTful APIs
Authentication
SOLID, YAGNI, KISS etc
Regular Expressions
GraphQL
Docker
C# (.NET)
Java (Grails, Spring, Play)
Go
Message Brokers
RabbitMQ
Apache Kafka
Caching
Memcached
Redis
Storage
Relational Databases
Oracle
PostgreSQL
MariaDB
MySQL
MSSQL
NoSQL Databases
Redis
MongoDB
Cassandra
RethinkDB
Couchbase
Solr
ElasticSearch
Sphinx
Search Engines
GOF Design Patterns
Architectural Patterns
Give DDD a shot
Learn different testing techniques
Up your Game further!
DevOps
Linux
Unix
Operating System
Terraform
CF Engine
Packer
SaltStack
AWS Cloud Formation
Automation
Chef
Ansible
Puppet
Cloud
Cloud Foundry
Rackspace
AWS
Heroku
Azure
Google Cloud Platform
Digitalocean
CI / CD
Jenkins
Travis
CircleCI
Drone
Bamboo
PagerDuty
Munin
Graphite
Prometheus
Zabbix
New Relic
Nagios
AppDynamics
Monitoring and Alerting
Papertrail
ELK
Graylog
Splunk
Log Management & Analysis
Containers
Docker
rkt
LXC
Kubernetes
Mesosphere
Mesos
Cluster Managers
Docker Swarm
Nomad
Web Servers
Apache
Nginx
Tomcat
IIS
Caddy
Differences and when to use what
Compiling apps from source (gcc, make and other related stuff)
Love for Terminal
Bash Scripts
Vim / Nano
Commands/Tools
Text Manipulation
awk, sed, grep, sort, uniq, cat, cut, echo, fmt, tr, nl, egrep, fgrep, wc ..etc
Process Monitoring
ps, top, htop, atop ..etc
System Performance
nmon, iostat, sar, vmstat ..etc
Network
nmap, tcpdump, ping, mtr, traceroute, airmon, airodump ..etc
OSI Model. TCP/IP/UDP Common ports
Knowledge about different file systems
Setting up a Reverse Proxy (Nginx ..)
Setting up caching Server (Squid, Nginx ..)
Setting up a load balancer (HAProxy, Nginx ..)
Setting up a firewall
TLS, STARTTLS, SSL, HTTPS, SCP, SSH, SFTP,
Postmortem analysis when something bad happens

OBJETIVOS

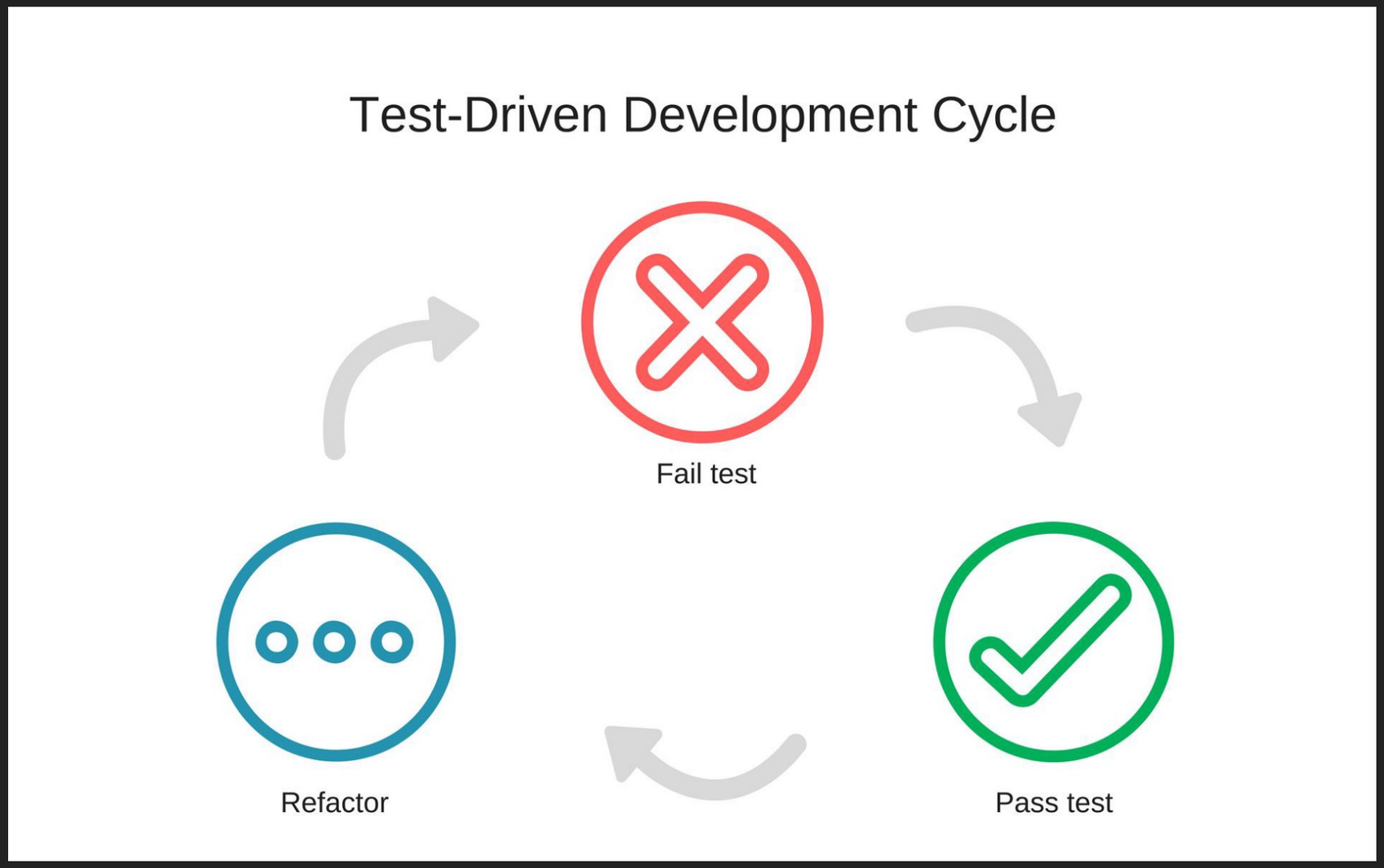
- COMPRE!
- COMPRE!
- COMPRE!
- BDD!

OBJETIVOS

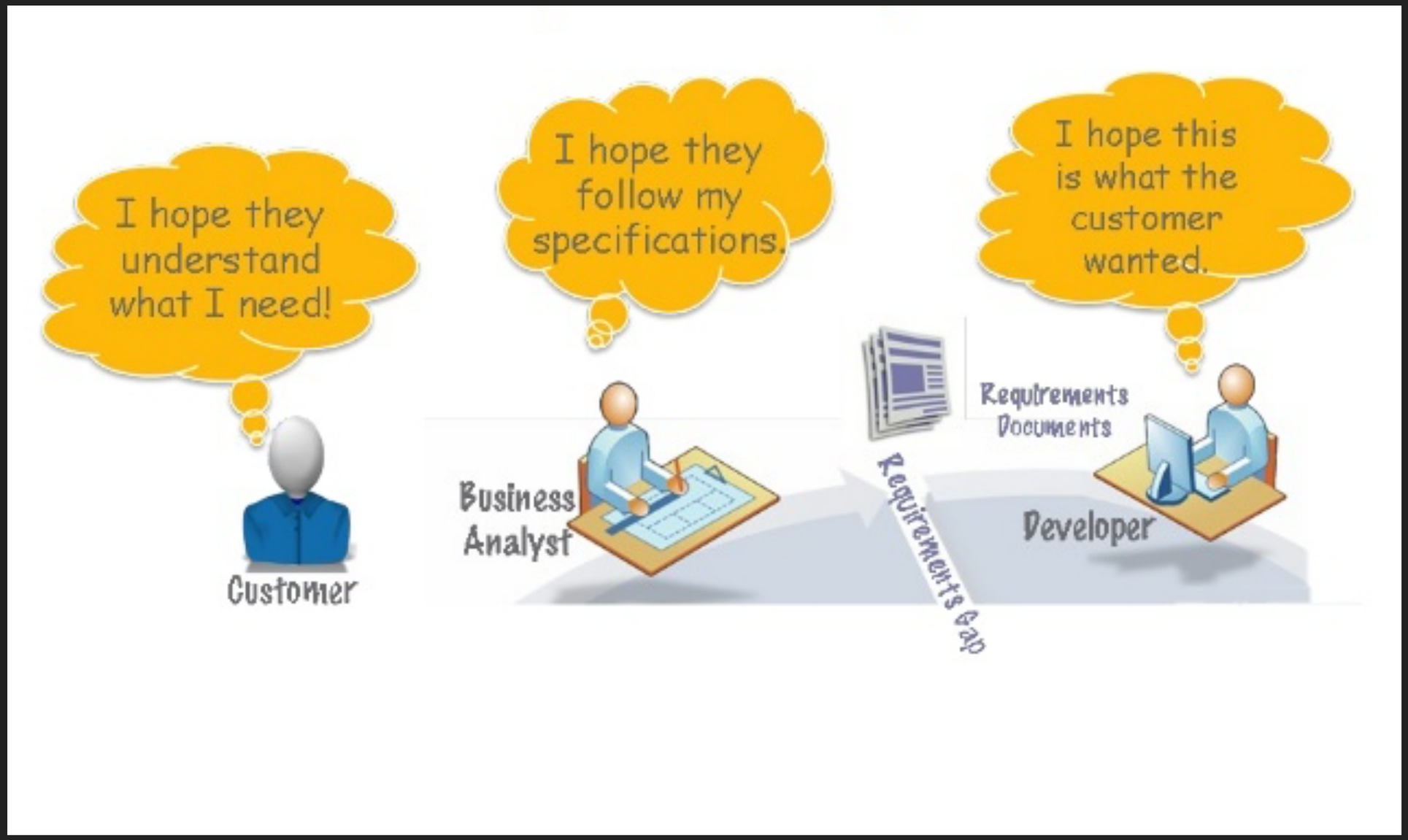
- Apresentar os conceitos
- Trocar experiências
- Grude4J

TEST DRIVEN DEVELOPMENT

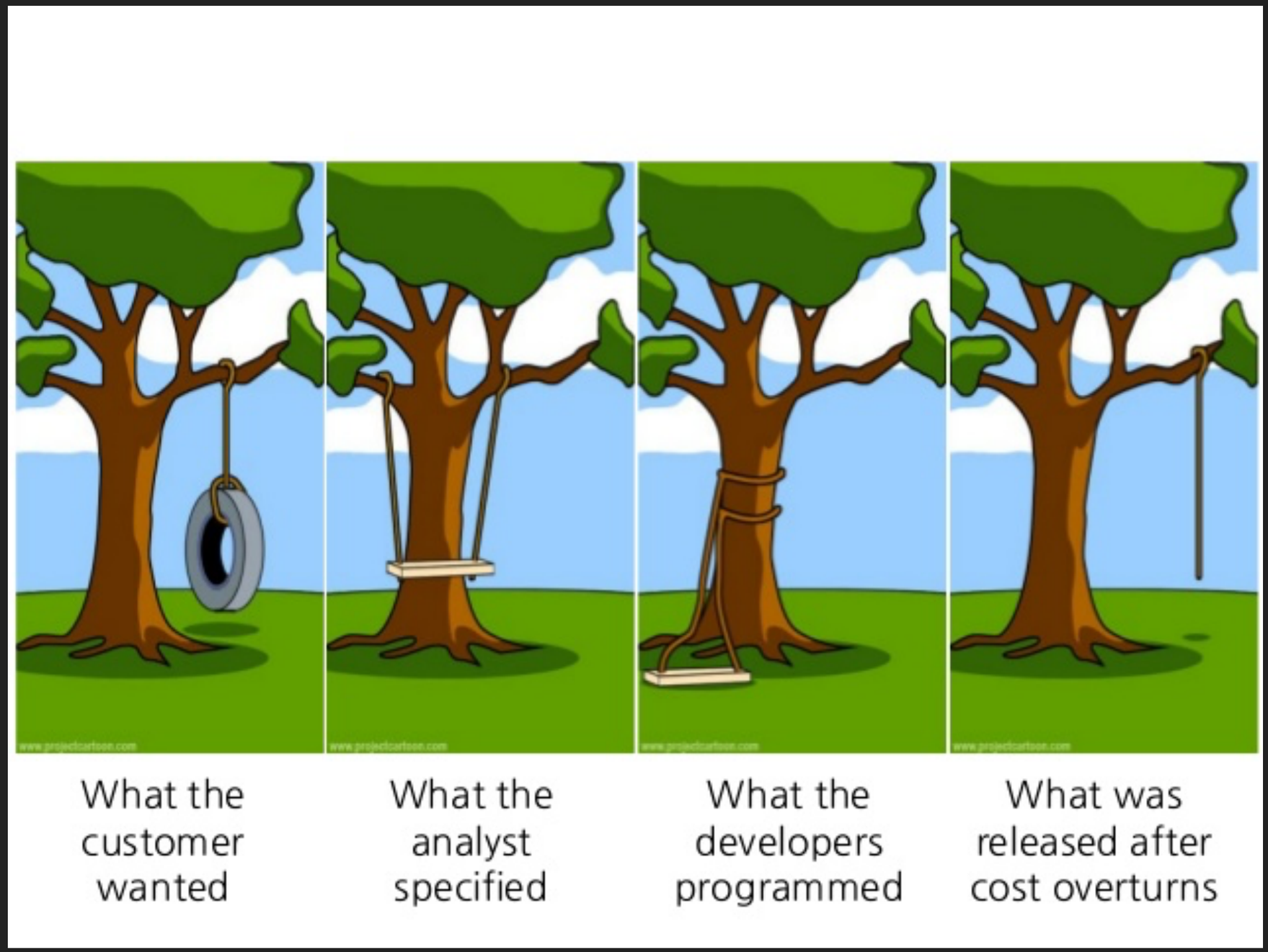
- Criado por Kent Beck, iniciado em 1999



O PROBLEMA


I hope they understand what I need!
I hope they follow my specifications.
I hope this is what the customer wanted.
Customer
Business Analyst
Requirements Documents
Developer
Requirements Gap

O PROBLEMA


What the customer wanted
What the analyst specified
What the developers programmed
What was released after cost overturns

O PROBLEMA

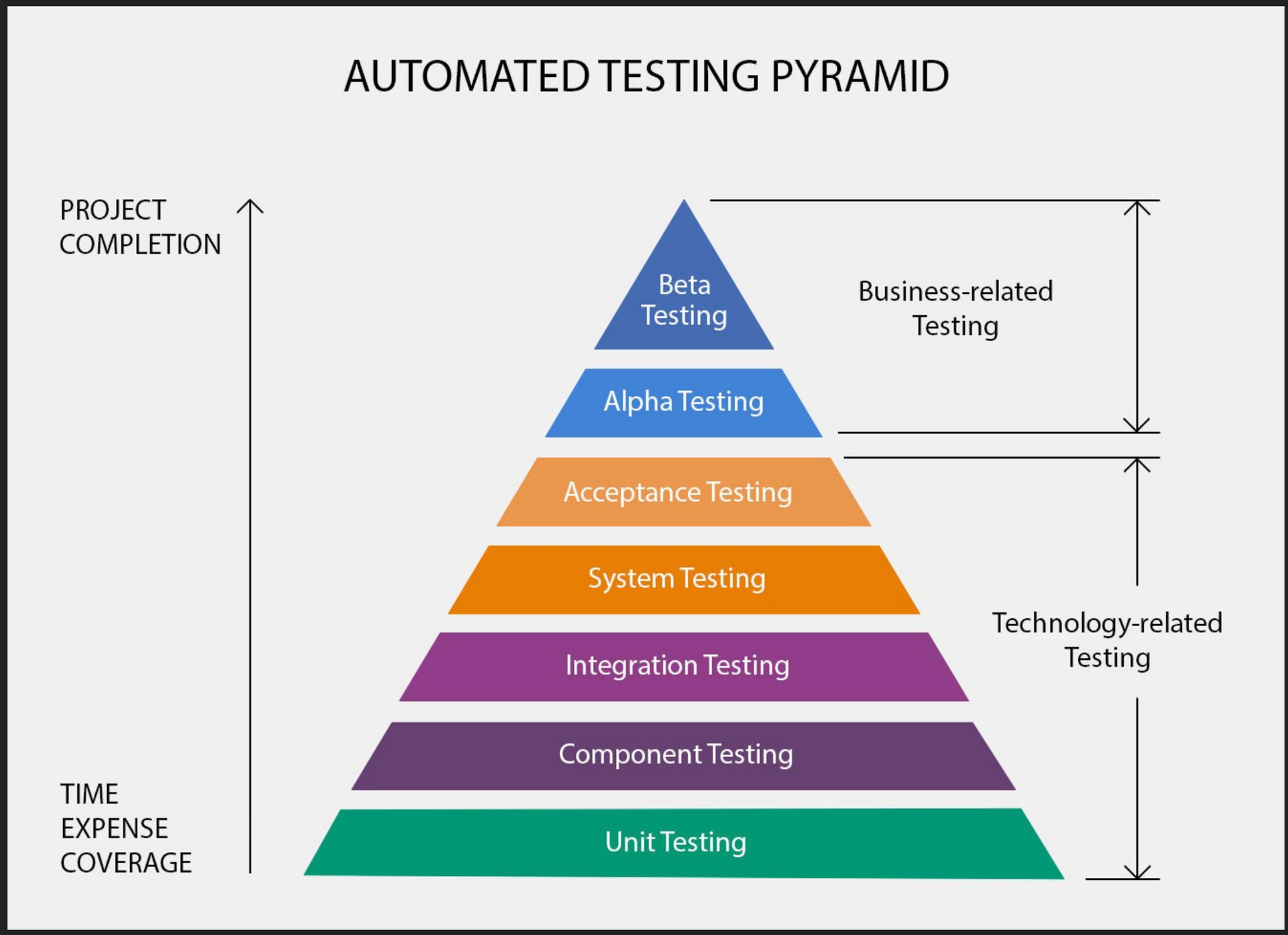

Defects found and remedied early in the software development life cycle reduce development costs and save time.
Cost per defect ($)
Requirements
Design
Development
Testing
Deployment
SDLC

TDD RESOLVE?

BDD

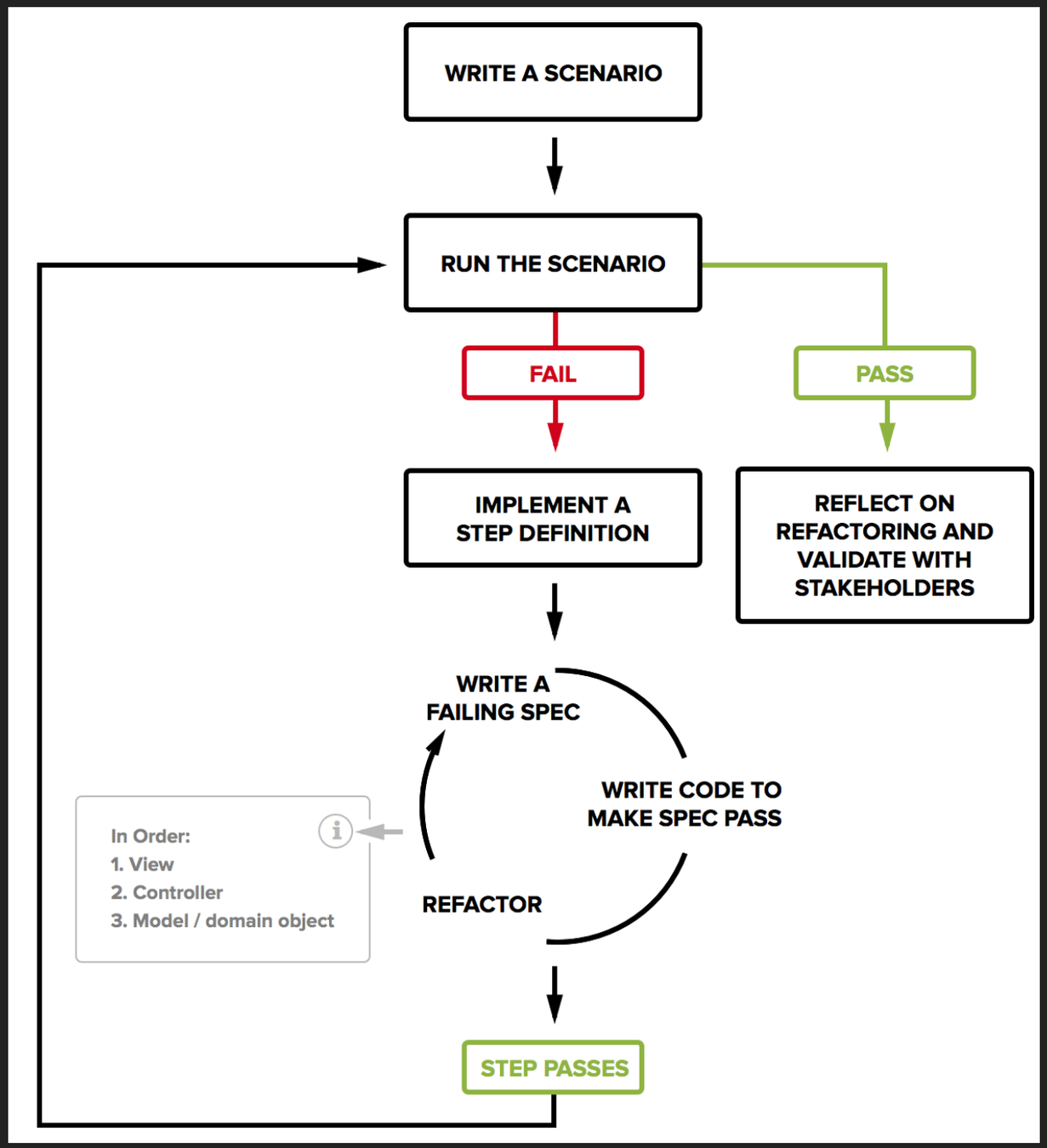
- “Evolução" do TDD
- Visa integrar Regras de negócio com linguagem de programação focando no comportamento do software.
- Criado por Dan North, em 2003
- “BDD é o uso de exemplos para discutir sobre como sua aplicação se comporta...e conversar sobre estes exemplos” Dan North, 2013

NÍVEIS DE TESTE DE SOFTWARE



PROCESSO DE DESENVOLVIMENTO COM BDD

- Especificação (Comportamento)
- Testes
- Desenvolvimento
- Refatoração



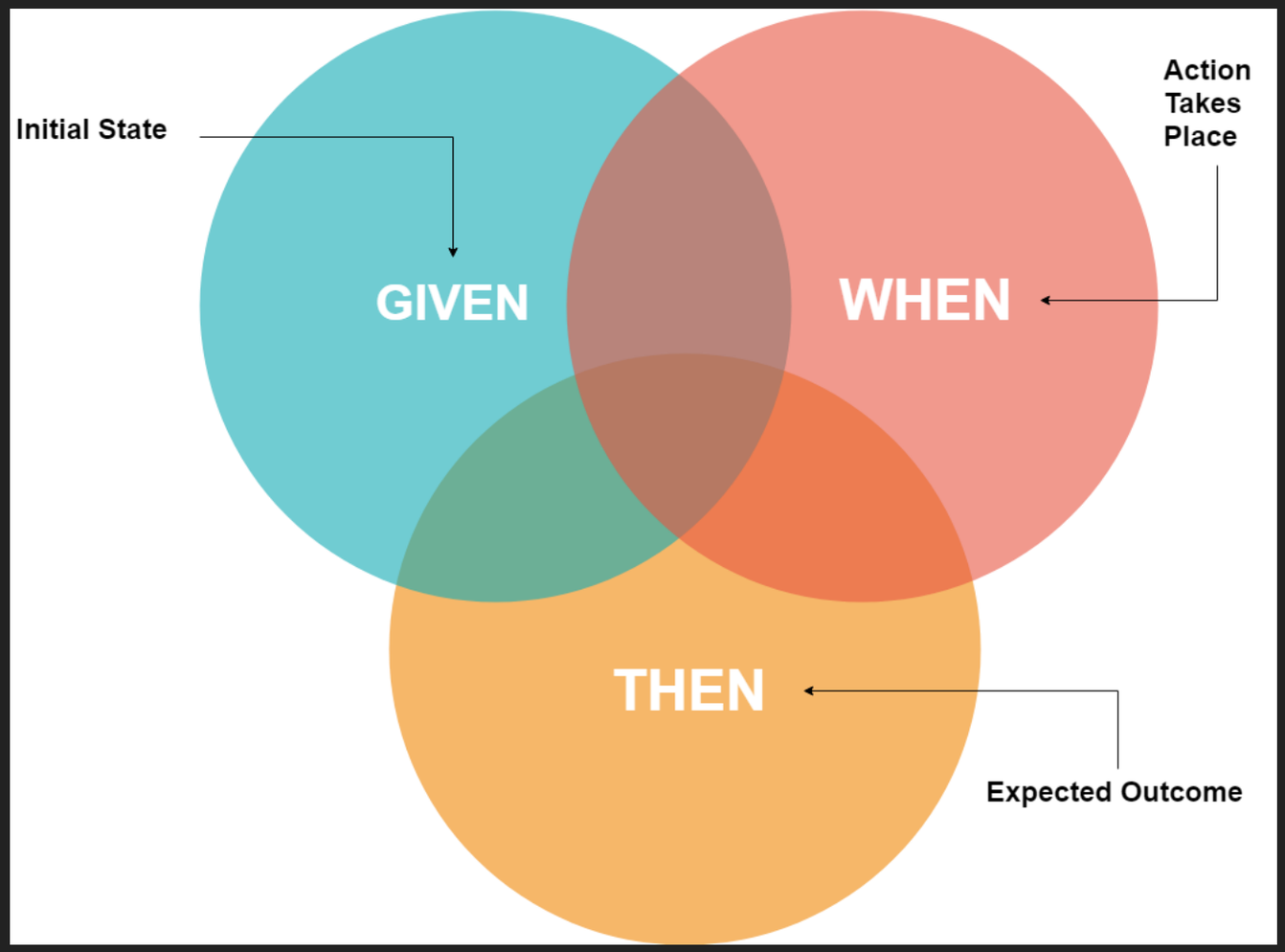
FEATURE

- Linguagem ubíqua (Gherkin)
- Suporta diversos idiomas
- “Rode”a feature para agilizar a criação dos passos

```
Feature: As a user I want to add two numbers to avoid human mistakes

Scenario: Add two whole numbers
    Given I insert the numbers "50" and "80" in my calculator
    When I click on "add" button
    Then my calculator displays the result "130"
```

FEATURE


Initial State
GIVEN
Action
Takes
Place
WHEN
THEN
Expected Outcome

DEFINIÇÃO DOS PASSOS

- Suporte a diferentes linguagens de programação

```java
public class Stepdefs {

        private Calculator calculator;

        @Given("I insert the numbers {string} and {string} in my calculator")
        public void insertNumbers(String number1, String number2) {
            calculator = new Calculator();
            calculator.setNumber1(Integer.parseInt(number1));
            calculator.setNumber2(Integer.parseInt(number2));
        }

        @When("I click on \"add\" button")
        public void i_click_on_button() {
                calculator.sum();
        }

        @Then("my calculator displays the result {string}")
        public void checkResult(String result) {
                assertTrue(result.equals(Integer.toString(calculator.getResult())));
        }
}
```

ISSO TÁ CERTO?

VANTAGENS DO BDD

- Melhora a comunicação entre equipes
- Antecipa riscos e bugs
- Pode servir como documentação

FRAMEWORKS JAVA

ALGUMAS FUNCIONALIDADES DO CUCUMBER

- Esquemas de Cenário

```
Scenario Outline: Price of a single item order
  Given I have not yet ordered anything
Given I have not yet ordered anything
When I go to the "<category>" category
And I select <item>
Then my current order total is <price>

Examples:
   | category   | item                  | price |
   | Sandwiches | a "Chicken Sandwich"  | $9    |
   | Dessert    | an "Oreo Cheesecake"  | $7    |
```

ALGUMAS FUNCIONALIDADES DO CUCUMBER

- Tags

```
@addItem
Feature: Adding an item to order
  I want to be able to add an item to a current order.

  @empty
  Scenario: Adding an item to an empty order
    Given I have not yet ordered anything
    When I go to the "Burgers" category
    And I select a "Cheeseburger"
    Then I have a new order
    And the order has 1 item in it

@price
Scenario Outline: Price of a single item order
    Given I have not yet ordered anything
    When I go to the "<category>" category
    And I select <item>
```

ALGUMAS FUNCIONALIDADES DO CUCUMBER

- Hooks
- Before, After

```
@Before
public void prepare(){
    // Set up something before each scenario
}
```

COMO FAZER EM JAVA

DISCUSSÃO