TEST DRIVEN DEVELOPMENT

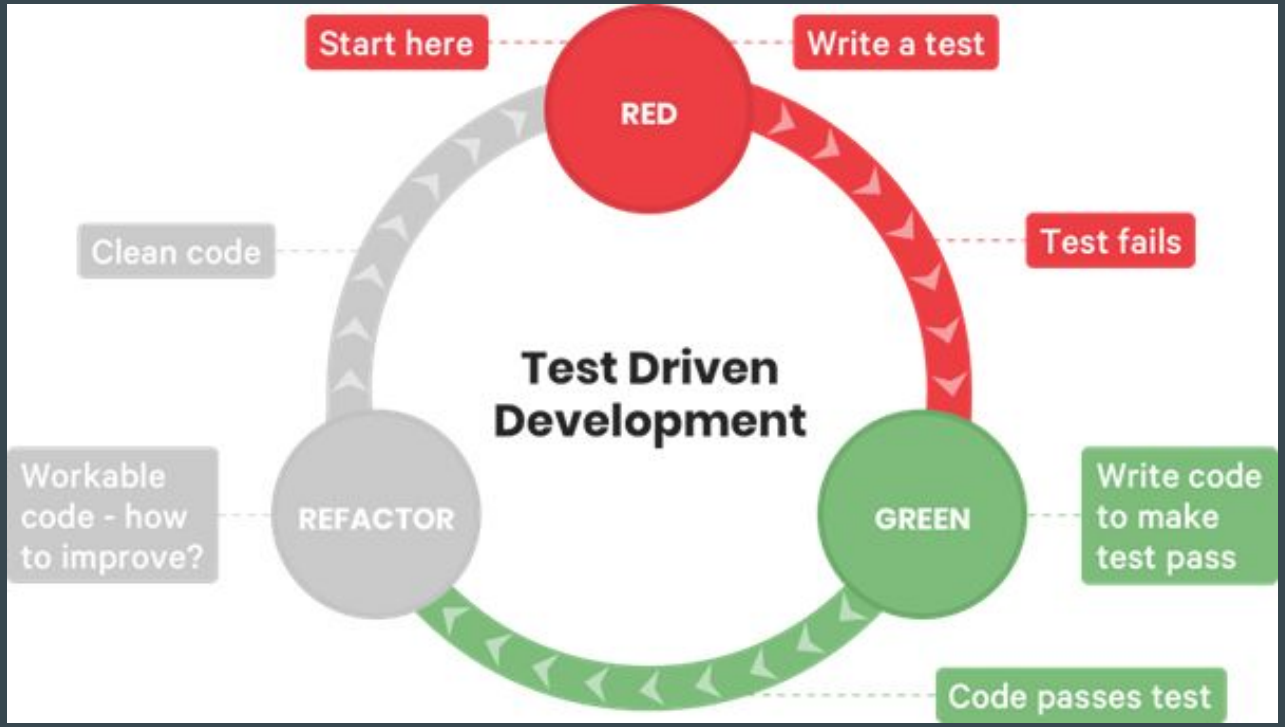
...

BY: APEKSHA JADHAV
BAVITRA
CHARITHA G
PALLAVI
SOMYA

WHY?

HOW?

BASIC CYCLE


Start here
Write a test
RED
Test fails
Clean code
Test Driven Development
Workable code - how to improve?
REFACTOR
GREEN
Write code to make test pass
Code passes test

FUNDAMENTAL PRINCIPLES

- Think about what you are trying to do
- Keep it simple
- Follow the TDD cycle
- Continuously make small and incremental changes
- No one can make a change that breaks the system
- Failures are addressed immediately

TOOLS

- JUnit for unit testing

  Unit testing is a modular approach followed in testing, single-single functionality or part of the code tested without waiting for the other code to be completed.

  -from the Article, *Unit testing techniques and best practices.*

- JMeter for Load / Performance testing

  Ability to load and performs test on many different applications/server/protocol types, some of them are listed:

  - Web HTTP, HTTPS(JAVA, PHP)
  - FTP
  - Database via JDBC

DEMO

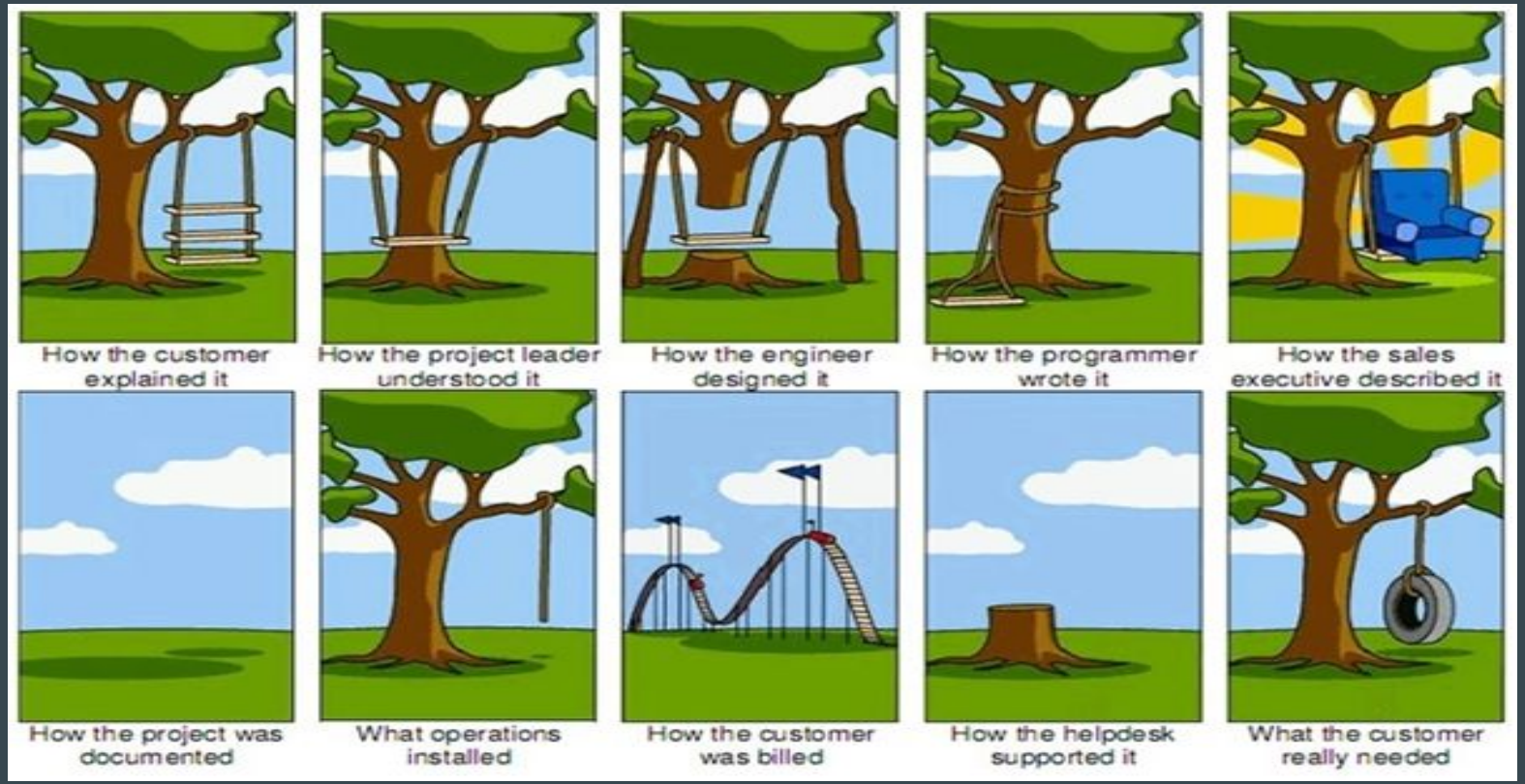
Benefits

- Much less debug time
- Fast feedback
- Code proven to meet requirement
- Development of high quality code
- Shorter development cycles
- Tests are documentation/detailed specification

Disadvantages

- **Big time investment**
- **Additional complexity**
- **Design impacts**
- **Continuous tweaking**

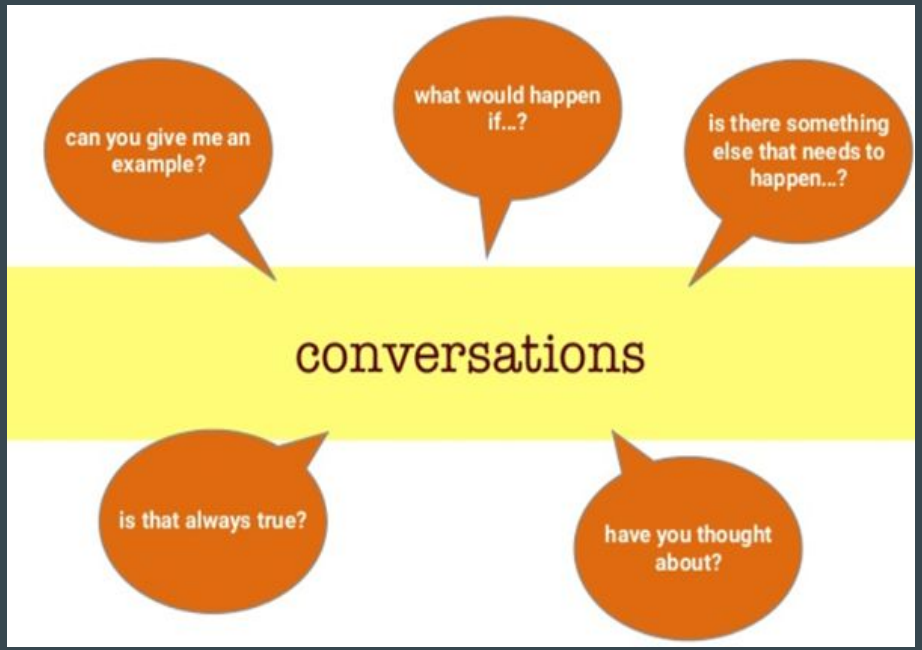
E-Commerce


How the customer explained it
How the project leader understood it
How the engineer designed it
How the programmer wrote it
How the sales executive described it
How the project was documented
What operations installed
How the customer was billed
How the helpdesk supported it
What the customer really needed

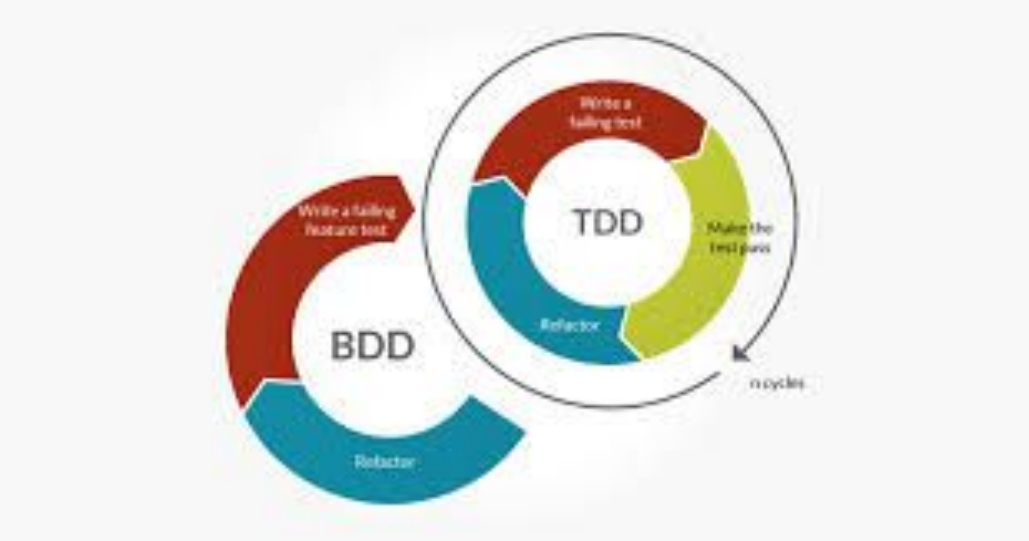
Behaviour driven development

Implementing the software by describing its behaviour from the perspective of its stakeholders and customers.

- **Given** a context
- **When** an event happens
- **Then** an outcome should occur



TDD vs BDD


Write a failing test
TDD
Make the test pass
Refactor
n cycles
Write a failing feature test
BDD
Refactor

References

https://en.wikipedia.org/wiki/Test-driven_development

https://www.browserstack.com/guide/tdd-vs-bdd-vs-atdd

https://www.guru99.com/test-driven-development.html

Thank you