NeverFail
Automated System Testing

A Project by:

Anthony Ofili, Aziz Alibrahim and Paloma Samaniego

Customer

Widget Computers Inc

- Users
  - Technicians
  - Hardware Engineers
- Administrative/Managers

Problem Diagnosis

The Problem

- Manual execution of tests
- Manual input of test results
- Need for human resources and availability
- Untrackable test history
- Email generation: human error
- Unscalable system

Overall high business costs on resources, budgeting and time utilization

The Problem

Overall high business costs on resources,

budgeting and time utilization

Proposed Solution

NeverFail Solution

- Automated test system (24/7)
- Tests for multiple machines
- Results stored in a central database
- Accessible test history on a web page

Business Benefits

- Reduces Manpower
- Decreases human presence in prototype testing
- Trackable errors and solutions
- Faster decision-making by managers
- Ensures data integrity
- Scalable project

Business Benefits

Overall higher cost-efficiencies

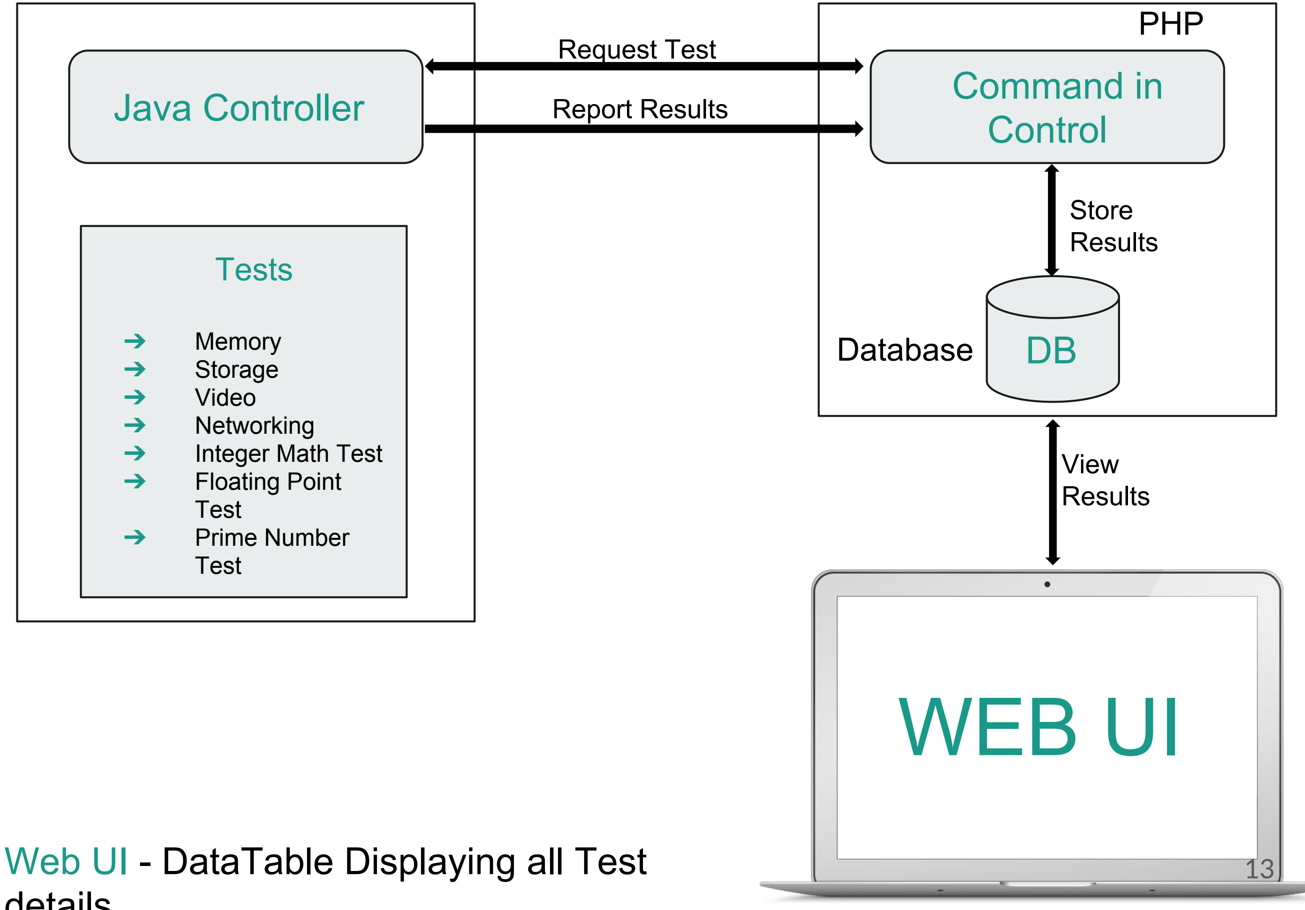
and cost-effectiveness

Construction Details

Building Blocks

- Inputs
    - Java controller
    - mySQL database
    - Python tests
    - PHP Command In Control
- Outputs
    - Web based GUI
- Workflow: Java controller requests tests from Command Control, tests will run, and ultimately test results will be posted to a web UI for the administrator to view.

System Diagram

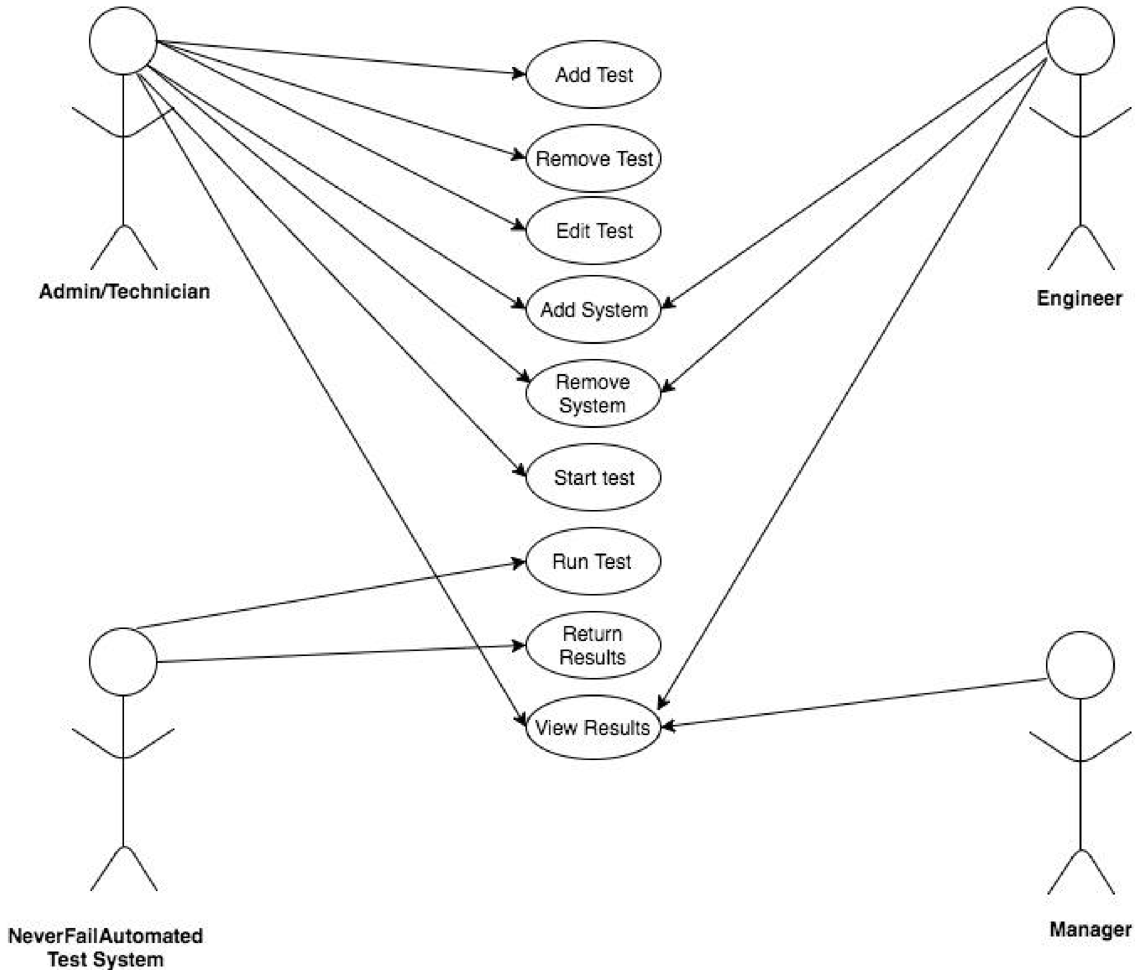


Web UI - DataTable Displaying all Test details

Use Cases

1. Adding tests
2. Executing tests
3. Analyzing results

Use Cases


Admin/Technician
Engineer
NeverFail Automated Test System
Manager
Add Test
Remove Test
Edit Test
Add System
Remove System
Start test
Run Test
Return Results
View Results

Initial Requirements

1. SUT requests tests from server
2. Tests will test major subsystems of a computer such as storage, networking and memory
3. SUT runs the tests and returns SUCCESS/FAIL
4. Test results stored in database
5. A web UI to display test results history
6. Persistent storage to store tests, results, and test client information
7. Automate the running of tests on multiple SUTs

Progress

1. Complete integration
2. Major subsystems check
3. Functional User Interface
4. Failure reporting
5. Ease of access

Technologies and Tools

MySQL®

Java™

php

JS

HTML 5

Bootstrap

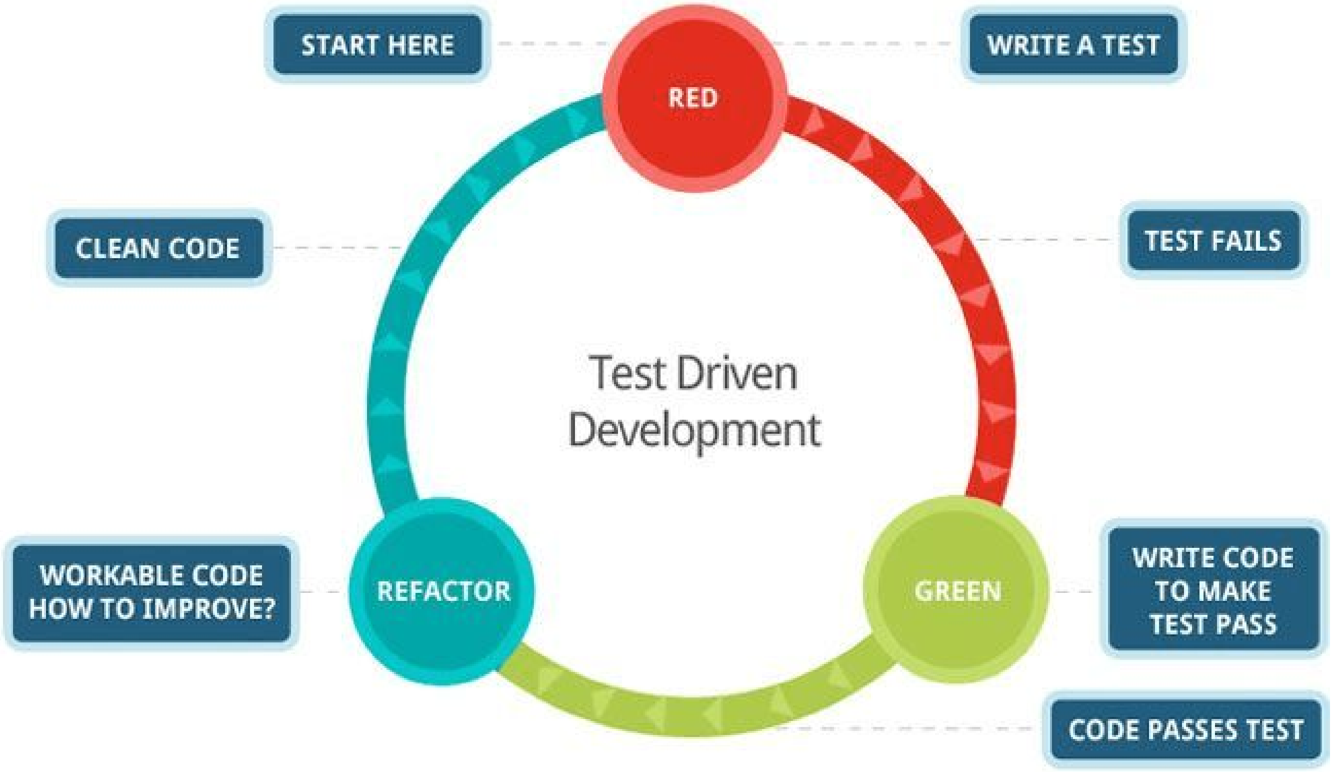
python™

Approach

Test-driven development

- Requirements set, test cases are written, code developed, run tests, refactor code, repeat
- Continuous customer interaction and input throughout the development process
- Early diagnosing of problems and solutions
- Complete understanding of the inner workings including corner cases.

Approach


START HERE
RED
WRITE A TEST
CLEAN CODE
TEST FAILS
Test Driven Development
WORKABLE CODE HOW TO IMPROVE?
REFACTOR
GREEN
WRITE CODE TO MAKE TEST PASS
CODE PASSES TEST

DEMO

Q&A