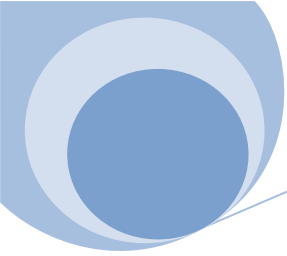




## Case Study:

*Apeiro Technologies testing services team helped client successfully implement test automation and significantly reduced test cycle time for their innovative approach to avail healthcare services.*



## Overview

Our client's product bridges the gap between patients and doctors by providing free public access to network of doctors to price their services without any mid-level companies in between. Patients can search for doctors and services and compare prices & services being offered. Patients can also avail healthcare services at discounted prices by purchasing Healthpons.

## Technology



- ~ Eclipse
- ~ Selenium Web driver
- ~ Java
- ~ TestNG

## Business Objectives & Challenges

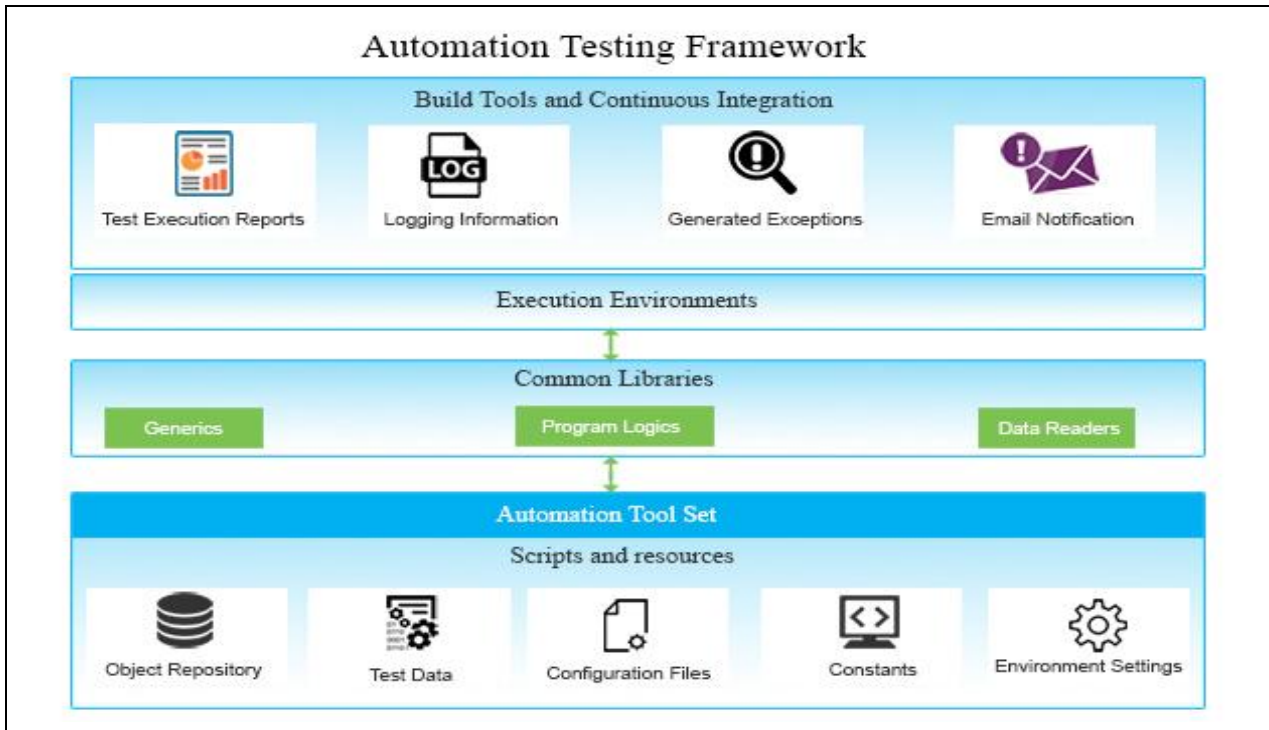
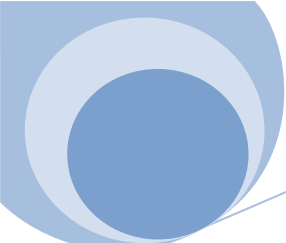
Our client's intent was to enhance quality of the application by automating their regression test scenarios requiring large data sets and different input types, to reduce the effort required for regression testing as well as standardizing test cases and test results.

They were also looking for test automation tool with robust framework which will allow easy modifications to test scenarios and maintainability of the test suite, along with the ability to automate test

## Technical Solution

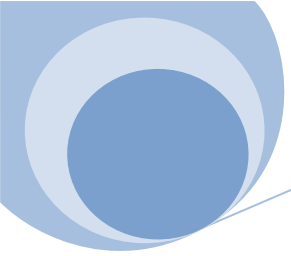
Careful analysis of the business requirements, and multiple discussions with the client resulted in shortlist of test automation tools along with Selenium TestNG framework.

After many brainstorming sessions with client and a feasibility study for test case automation, we concluded that Selenium with TestNG framework is the best tool for client's requirements. We tested complex functionalities of the application for correctness, reliability and effectiveness of tool; and results were positive, we also found that many crucial functionalities of the application can be automated using Selenium



## Framework

- ◆ **Object Repository:** This file contains set of locator types for various web elements.
- ◆ **Test Data:** This file maintains test data for testing different repetitive scenarios.
- ◆ **Configuration File:** This file contains all the information related to the application URL, specific information related to various browsers. Generally this file is used to maintain all the static information throughout the framework.
- ◆ **Generics/ Program logics/ Readers:** Stores common functions that are used across the framework
- ◆ **Build tools and Continuous Integration:** External tools that enhance the capabilities of the frameworks like generating reports, email functionality and logging information.



## Our Solution

- ◆ Building the framework and maintaining scripts was a challenging task as there were frequent GUI and functionality changes – Our experience with Agile development methodology greatly solved the problem.
- ◆ Executing test suite with defined UI locators used to fail randomly in IE browser - We created a separate property config file for IE.
- ◆ CSS locators rather than XPATH to increase the speed of execution in IE browser.
- ◆ We implemented dynamic wait functions and a global level variable to handle the variation in application response time.
- ◆ Extensively supported major releases to make sure the application can be launched on time and with quality.
- ◆ Handling dynamic locators that are generated at runtime, such elements were identified by going through hierarchy and using exact web element text, exact class name or getting unique attributes.

## *Our Approach : Plan-Do-Check-Adjust (PDCA)*

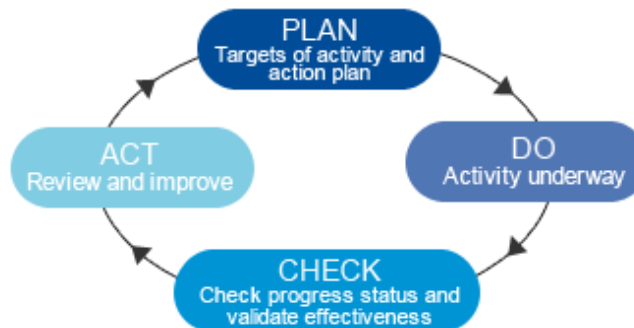
To accomplish the task of creating strong framework and to handle script modifications we followed PDCA model of management which is one of the best practice suited for agile environments.

### **Plan**

To design and architect the TestNG Framework, business requirements like regression test cases, complex test scenarios, minimum number of test cases to be executed, test data, standards to be followed for script designing and types of browsers to be tested were carefully analyzed.

### **Do**

With the information gathered during planning step, first we started developing framework for few scripts then we set up the execution environment, installed all the needed software's, and created folder structure, property files, and configuration files.



### **Check**

We tested the framework by executing few scripts from all the scenarios for regression test cases to evaluate the status of the framework.

### **Adjust**

Based on the results in the check step, we improved the framework by standardizing the features of the framework; this process continued till the goal of developing a strong framework and designing scripts based on the framework was achieved.

## Key Benefits of Using Selenium Webdriver in Automating Test Cases



- 1 Test Automation resulted in identifying 120+ defects during regression testing.
- 2 Overall there was reduction in testing time by more than 2 weeks per sprint.
- 3 Reduced costs by using our unique global delivery methodology employing a mix of onshore and offshore resources.
- 4 Our team of experts with years of testing experience and expertise supported client right from framework/technology selection, to hosting, and launch of the application
- 5 Quick and responsive team members were easily reachable for quick issue resolution.
- 6 Testing team for UAT created large test data sets by using automation scripts

### Snap Shot: of the key benefits delivered

Automation Test suite	No of Scenarios Tested	Manual Effort(Days)	Automation Effort(Days)	Saved Effort (Days)
Patient Module	50	8	3	5
Doctors Module	70	11	5	6
Creating Healthpons Module	25	5	2	3
Book Appointments	35	7	3	4
Purchase Healthpons	20	4	1	3
Data Reading Scripts/POI API's	20	4	1	3
Appointment Date Verification scenarios	15	3	1	2
Complex calculation Verification scenarios	35	7	2	5
Admin Module Verification scenarios	80	16	2	14
Remaining Functional Testing	200	30	6	24
UAT	550	80	30	50