Test Approach for Buythisspace.com.au

The following components will comprise of test strategy document to validate the portal "http://buythisspace.com.au/"

Prerequisite to start testing :
1. Requirements has been identified and added to backlog
2. All the required addons has been identified
3. Scrum team has been established
4. Delivery date has been discussed and agreed with Product Owner
5. Test platform has been created separate from Production

When to test:-

- Website under test (AUT) need to satisfy the functionality agreed on each deployment and release
- AUT need to secure enough that no vulnerable element can destroy the content of the website
- Performance of AUT need to be fast enough for users to use it e.g. response time of 3sec for each page load
- Confidence of your visitors is the value that you need to maintain while testing AUT or during any other updates
- Site would be easy to access and accessibility of the site has been validated for disable users (Vision Disability, Physical Disability, Cognitive disability, Literacy Disability, Hearing Disability)
- Establish an automated suite to validate all of the above in DevOps model for iterative releases
- Split test analysis for better user experience and increasing site metric

Process/strategy of initiating this test:-

1. Validate the versions of WordPress and its plugins are latest
2. Create virtual user profile in JMeter to test the application under 500 Concurrent user load
3. Run a security scan using Kali Linux and WPScan
4. Run a
5. Validate all the addons being used are latest and project team is aware of implementation steps to achieve the desired features i.e. In any CMS site dev team is limited with features and configurations available in addon and in that case it is vital to understand the implementation steps in test before putting it on production. Vulnerable components being used in BuyThisSpace.com.au
    a. WordPress Version: 2.7.1
    b. WordPress 2.0 - 2.7.1 admin.php Module Configuration Security Bypass
    c. WordPress 2.5 - 3.3.1 XSS in swfupload

d. WordPress 1.5.1 - 3.5 XMLRPC Pingback API Internal/External Port Scanning
e. WordPress 1.5.1 - 3.5 XMLRPC pingback additional issues
f. WordPress 2.0 - 3.0.1 wp-includes/comment.php Bypass Spam Restrictions
g. WordPress 2.0 - 3.0.1 Multiple Cross-Site Scripting (XSS) in request_filesystem_credentials()
h. WordPress 2.0 - 3.0.1 Cross-Site Scripting (XSS) in wp-admin/plugins.php
i. WordPress 2.0 - 3.0.1 wp-includes/capabilities.php Remote Authenticated Administrator Delete Action Bypass
j. WordPress 2.0 - 3.0 Remote Authenticated Administrator Add Action Bypass
k. WordPress 2.0.3 - 3.9.1 (except 3.7.4 / 3.8.4) CSRF Token Brute Forcing
l. WordPress <= 4.0 - Long Password Denial of Service (DoS)
m. WordPress <= 4.0 - Server Side Request Forgery (SSRF)
n. WordPress <= 4.4.2 - SSRF Bypass using Octal & Hexedecimal IP addresses
o. WordPress 2.6.0-4.5.2 - Unauthorized Category Removal from Post
p. WordPress 2.5-4.6 - Authenticated Stored Cross-Site Scripting via Image Filename
q. WordPress <= 4.7 - Post via Email Checks mail.example.com by Default
r. WordPress 2.3-4.8.3 - Host Header Injection in Password Reset
s. WordPress 2.7.0-4.7.4 - Insufficient Redirect Validation
t. WordPress 2.5.0-4.7.4 - Post Meta Data Values Improper Handling in XML-RPC
u. WordPress 2.5.0-4.7.4 - Filesystem Credentials Dialog CSRF
v. WordPress 2.3.0-4.8.1 - $wpdb->prepare() potential SQL Injection
w. WordPress 2.3.0-4.7.4 - Authenticated SQL injection
x. WordPress <= 4.8.2 - $wpdb->prepare() Weakness
y. WordPress 1.5.0-4.9 - RSS and Atom Feed Escaping
z. WordPress <= 4.9.4 - Application Denial of Service (DoS) (unpatched)

6. Plugins and Themes:
    a. DANDO_OOH_BTS - V4.1
    b. ADVANCED-ACCESS-MANAGER - V5.0.4 - New version available
    c. DURACELLTOMI-GOOGLE-TAG-MANAGER - V1.7.2
    d. GEO-MY-WP - V2.7.1
    e. GMW-GLOBAL-MAPS - V1.1.2
    f. GMW-PREMIUM-SETTINGS
    g. GRAVITYFORMS - vulnerable

i. **Gravity Forms <= 1.8.19 - Arbitrary File Upload fixed in 1.8.20**
ii. **Gravity Forms 1.8 <= 1.9.3.5 - Authenticated Blind SQL Injection fixed in 1.9.3.6**
iii. **Gravity Forms <= 1.9.6 - Cross-Site Scripting (XSS) fixed in 1.9.7**
iv. **Gravity Forms <= 1.9.15.11 - Authenticated Reflected Cross-Site Scripting (XSS) fixed in 1.9.16**
v. **Gravity Forms <= 2.0.6.5 - Authenticated Blind Cross-Site Scripting (XSS) fixed in 2.0.7**

h. **WIDGET-OPTIONS - V1.0- New version available**
i. **WORDPRESS-SEO - V6.1.1- New version available**

7. Information Leakage :
    a. robots.txt available under: 'http://buythisspace.com.au/robots.txt'
    b. Interesting entry from robots.txt: http://buythisspace.com.au/wp-admin/admin-ajax.php
    c. The WordPress 'http://buythisspace.com.au/readme.html' file exists exposing a version number
    d. Interesting header: LINK: ; rel="https://api.w.org/"
    e. Interesting header: LINK: ; rel=shortlink
    f. Interesting header: SERVER: Apache/2.4.18 (Ubuntu)
    g. Interesting header: SET-COOKIE: wfvt_2378251233=5aa8a0c8d905f; expires=Wed, 14-Mar-2018 04:40:48 GMT; Max-Age=1800; path=/; HttpOnly
    h. This site has 'Must Use Plugins' (http://codex.wordpress.org/Must_Use_Plugins)
    i. XML-RPC Interface available under: http://buythisspace.com.au/xmlrpc.php
8. Identification of high priority items:-
    a. Identify a list of themes or items that need to be tested as important part of deliveries
    b. Create a list of URLs that needs to tested after the testing have passed and also the list of URLs that need to be checked as part of legacy
    c. Create a list of a business flow that comprises of user experience and related to a journey for example
        i. User navigates to this site ““http://buythisspace.com.au/””
        ii. Parametrize user search for area e.g. TestData file having multiple search location and framework validates the results for each parameter “Epping” within 1 km range
        iii. From the available results,select a place and use Enquire now button to proceed
        iv. Select the environment where you want to test the application e.g. Stage, Test of Live environment
        v. Create a control run from your existing application and use this as baseline to compare later results this may include screenshots or automated test if any

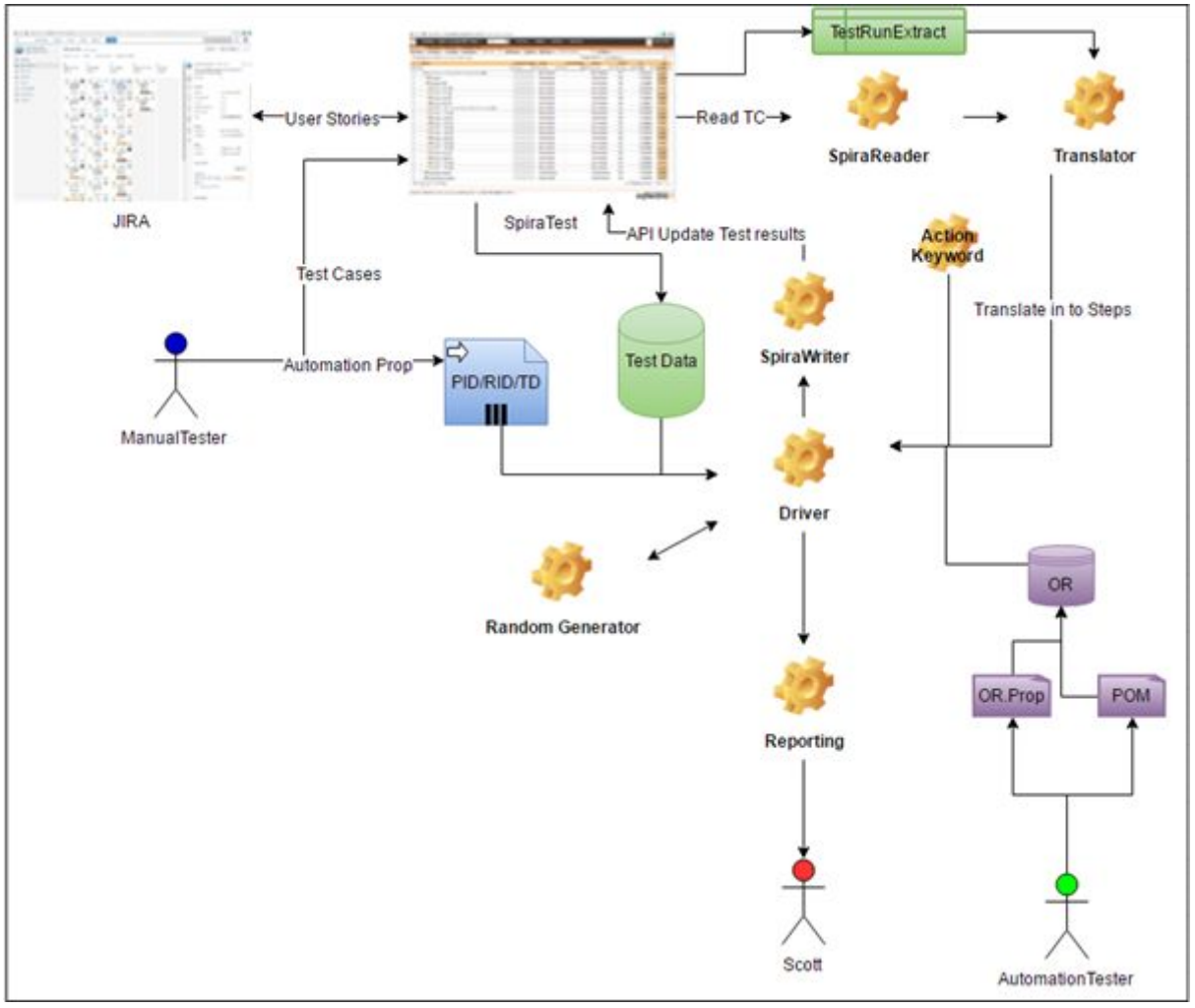
vi. Make sure to visit all your operating browsers and mobile devices to create control runs eg. Chrome or iphone

vii. Deploy your site with latest fixes now and rerun all your planned test and create a test run for the same browsers and devices with same automated test suite which you used for your control runs

viii. Compare the results of control vs test and verify the results of your test and fixes

D. Walkthrough all your steps given above from A. to C. and verify the results are accurate or as expected with control runs.

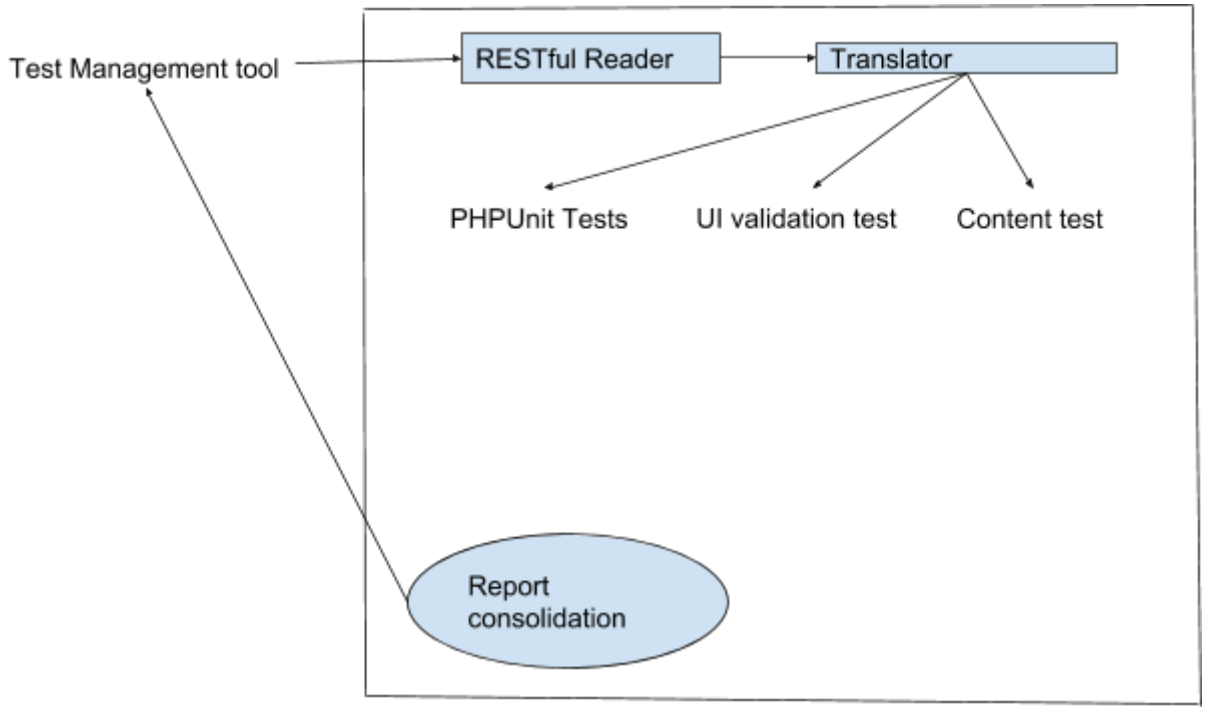
Automated test: One of my recently developed framework that works on Behaviour driven keyword test allows manual users to write test cases without changing the code and minimum code maintenance as test cases are written in JIRA not with in the code.



Proposed Test Framework:

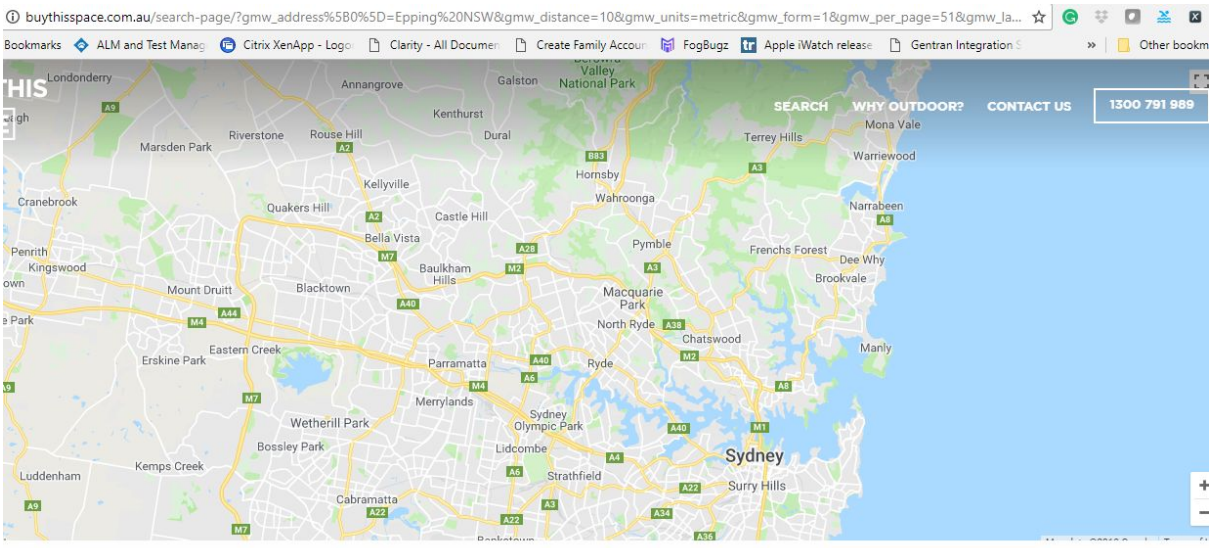
1. Maintain test cases and test plan within test management tool
2. Framework developed in Python/Java to read test cases from test management tool and translate them in executable based on desired actions

3. Execute all test using different tools such as PHPUnit, SoapUI, Selenium, WPScan, JMeter, Screenshot creation tools, SikUli
4. Making the site high performing website with fast loading speeds. Implement suite of free online tools and WordPress plugins find the trouble spots and fix them without too much work. How do you keep your site lean and running fast



Sample Defects :-

1. Search for Epping with 10 km of search then the results appear with a map at the top , in this map there is no indication of selected region and no billboards are displayed here at this map. Ideally this should draw or highlight



2. When any area is selected then the map should clearly show the business name while its shown as blank currently