

SUBJECT

INFORMATION SYSTEMS PROJECT MANAGEMENT

**TOPIC: SESSION 10: CASE STUDY My Hands-On Experience
Software Testing**

**CASE STUDY:My Hands-On Experience
Software Testing**

SESSION 10

CASE STUDY

My Hands-On Experience

Software Testing

Well-versed with formal and exploratory software testing approaches in traditional and iterative software development models.

- Formal and exploratory testing of web applications, local desktop applications, SDK packages
- Used various software test design techniques which are traditionally called as black-box/white-box, static/dynamic testing techniques
- Known in the testing industry for challenging some well established thoughts from the renowned experts like Schools of Testing, Testing vs Checking, Formal vs Exploratory testing etc.
- Developed an improved parser-friendly text session report format for exploratory testing using Session based test management. Developed Python code to parse all session reports periodically to generate consolidated PDF reports for the team.
- Introduced Test Doubles concept to system testing to increase test coverage
- Employed All-pairs technique to API testing and event capturing testing
- Worked on Database migration testing from hierarchial (IMSDB) to RDBMS (DB2). Was chosen by management to take up training for the complete development team on testing of the integrated system.
- White box testing using code coverage tools

Security Testing

His experiments with Python and fuzzing made him the first to present on the subject of fuzzing at any software testing conference, voted as the Best Innovative Paper Award by audience. He moved on to present extended tutorials on the subject with demonstrations of exploits using Python.

- Conducted web security testing for multiple clients and reported critical security bugs not found by automated scanners.

- Well-versed with web security testing – exploration with ready-made tools (proxies/browser extensions etc.) and customized tools written in Python
- A File fuzzing rig for scanning engine
- Developed a PoC for fuzzing the command line interface of a software
- Did a security evaluation of usage of system resources like pipes for a middle tier local software and found critical vulnerabilities.

Performance Testing

Done web performance evaluation for American Airlines, Brown University, Riyadh Bank and many others. Presented a lightning note on how to convert all functional tests into heuristic performance tests at Google Test Automation conference (GTAC-2010).

Also, dealt with non-web performance testing (i.e. desktop GUI applications or SDK packages), for example performance analysis of boot time, application activity, testing via API, memory footprint, encryption cycles performance etc.

- Undertook 20 performance testing assignments with target user load ranging from 500 to 10000 users, employing various tools like LoadRunner, WebLoad, AppMeter and OpenSTA and a test environment up to 625 machines.
- Developed the Online Reporting Portal for Performance Testing in Perl
- Point of Contact for Tool Evaluation and Business Proposals in Performance Testing Vertical of AppLabs.
- Was nominated for a discussion with Microsoft's performance team on limitations of current market tools and desirable features in next-generation tools.
- Developed Siebel parser to overcome LoadRunner's limitation to deal with Arabic data.
- Submitted case study to RadView to facilitate their tool modification to support SiteMinder. WebLoad 7.3 now supports SiteMinder.
- Memory Footprint Analysis Automation using Python, API testing tool and PerfMon
- Performance Evaluation of Encryption formats for signature files
- Automated Outlier analysis in performance data

Interviewing and assessment

- Evaluated hundreds of candidates so far.

- The target roles include freshers, test engineers, developers, leads, managers, technical specialists/architects.
- I have a track record of 5 mis-hires, from which I have learned and improved my assessment skills. I know this, because I retrospect.
- Introduced discussion based interviewing rather than “What is...” questions
- Strong focus on demonstration skills rather than recalling skills
- Followed “learn and demonstrate” methodology for interviewing freshers, where I teach a technique during the interview and then request for demonstration from the freshers, as they are mostly uneducated in terms of testing.
- Interviewees have often found my interviews intriguing and useful, and noted that it is a welcome change.

Test Automation

Hands-on experience of designing and implementing multiple test automation solutions. The automation solutions range from specific implementations for performance testing, white box testing, fuzzing, auto-generated empirical analysis of quality of builds from automated tests, distributed testing and the likes to generic object oriented test frameworks that are used by multiple teams including developers and testers.

I have given various unique ideas to the test automation world like Test Encapsulation and Test Automation Support Systems with Layered Architecture.

Some snapshots of his work in test automation include (performance/security work related information is not included here, although the same ofcourse includes automation):

- Developed Python based framework for validation of Microsoft SSRS database w.r.t the back-end.
- Developed JMeter Web Based Reporting Portal
- TERMS framework, developed for McAfee, used by multiple teams in McAfee Labs
- Anti-Malware Core TMS: Test Automation framework developed for McAfee's Anti-Malware Core Team
- Wrote an event generation tool for various permutations of Windows API calls got via all-pairs in Python for testing filter kernel drivers.
- Revamped an anti-malware engine performance testing rig
- Introduced the idea of Test Doubles for system level test automation (along with white box tests)
- Developed module for scripted ESX virtual infrastructure management
- Developed a PoC for HTTP Proxy in Python

- Developed a PoC for customer wxPython browser with navigation hooks for automatic script generation
- Developed a test automation framework for a local e-learning application in Auto-It.
- (Many years back) Web automation at GUI layer for a dynamic insurance claims application using SilkTest (no record-playback)

Agile Testing

- Worked for 4+ years in projects employing Scrum
- One year experience as a ScrumMaster
- The semi-agile (mini-waterfallish) environment helped in identifying the core agile values that are important, how Agile can be tailored into more acceptable format for the newbies
- Introduced Test automation upfront in agile testing using generic object oriented test automation framework for developers and testers
- Experimented with Test Driven Development in Python
- Experimented with Python and Fitness working together
- Reviewed user stories (problem statement) from coverage perspective for various software attributes
- Developed testing tracker from user stories to the white-box tests (component tests)

Python Programming

- 6+ years experience in Python programming
- Employed Python for development (as the sole developed and architect) of various test automation solutions, frameworks, tools and test automation support system
- Development a basic web application using Django framework
- Using the HTTP request libraries in the core module, writing a custom module, developing an HTTP proxy from scratch at TCP layer
- Development of fuzzing frameworks employing generation and mutation
- Development of GUI user interface using wxPython for a performance testing tool with a tabs for recording/browser/editor.
- Distributed testing using Push (at TCP layer) and Pull (HTTP layer) mechanisms
- Object serialization (Pickling) for test configuration format
- Basic XML parsing coupled with tools like PICT (from Microsoft) or custom code

- Using ctypes on top of VIX dll for virtual machine management via VCenter or host.
- SBTM (Session Based Test Management) parser and reporter
- Python with raw Windows API for enhanced calls to file handling operations
- Extensive experience in handling nested data structures
- Implemented various design patterns like singleton pattern, factory pattern, decorator pattern, command pattern etc. in test automation frameworks.
- Designed a 50+ table database using SQLite for test automation framework, developed comprehensive database layer using Python code. Used the advanced RowFactory instead of the default array returning mode.

Case Study: Exploratory Product Comparison of FotoBounce(TM)

In my role as Chief Testing Officer at **Test Mile Software Testing Pvt Ltd**, I recently consulted for an exploratory product comparison testing for **FotoBounce(TM)** along with a team of Test Milers. As a company, Test Mile is known for being a skilled testing team with a strong knowledge of coding. I have been advocating programming as an essential skill for testers. This is first time that Test Mile handled a project that did not involve programming, and I hope that the report shared herein once again reinforces the message that knowledge of coding does not hamper your exploration skills rather it is a valuable asset.

Products compared

FotoBounce(TM) Desktop application was compared in terms of its face recognition capabilities with **Google Picasa(TM)**, **Apple iPhoto(TM)** and **Microsoft Photo Gallery(TM)**. (All the names mentioned here are registered trademarks of the respective companies.)

Methodology

- It was undertaken as a 2-week project by Test Mile.
- A photo-set of 400 photographs spread across 8 events/groups was considered
- The sequence of operations in terms of adding groups, tagging faces etc. was consistently done across all products
- The steps were executed by a single engineer, in multiple uninterrupted sessions with extensive note taking
- Comparison parameters were identified in collaboration with FotoBounce team

- The exploratory session reports were later consolidated and the report was furnished

On the Degree of Independence

- Right from the first discussion, FotoBounce team was on the same page as Test Mile in terms of independence of testing.
- FotoBounce team had not control over which photographs are picked up, the final list of comparison metrics and the way Test Mile planned to do it.
- I really appreciate FotoBounce team's openness to publish Test Mile report as such, without removing the sections which point out the areas wherein their product needs to improve. It is a sign of great professionalism.