SUCCESS STORY FOR QA FOR WEB APPS DEVELOPMENT

BACKGROUND

To successfully develop and deploy an Enterprise Scale Web Application to serve hundreds of thousands of customers the client required stringent quality control. Due to a very large consumer base there was the need of an intricate Quality Assurance (QA) and Testing methodology to enable checks to be performed on the functionality of the application and eventually rectify errors. Setting up an efficient QA Team while battling a rough time constraint had become a significant roadblock for the client. As a result the client decided to rely on the Enosis' Testing and QA team to ensure thorough execution of a variety of tests that would lead to successful error elimination prior to the final implementation of the solution. Enosis utilized the extensive experience of its enterprise-wide Quality Assurance & Testing team in the field of Web Application testing to set up a concrete foundation for testing the solution under a wide variety of conditions that are bound to arise due to the various behavioral patterns that exist within large consumer base.

THE CLIENT

SpoonByte is a US based "software on demand" service provider offering iPhone & Web app that allows restaurants to create promotions (e.g., meal discounts, menu specials, or events) to increase revenue. It's the first app that serves the food industry by focusing on Match-Based Deal Dissemination for the physical marketplace.
SpoonByte intelligently sends each promotion only to consumers that are near the restaurant and are predicted to act on the promotion and visit the restaurant.

THE APPLICATION

Small and mid-sized restaurants lack customers even during peak dinner time and consumers are unable to find the appropriate deals at the right time. To address the needs of the food service industry the client collaborated with Enosis to develop a customized Business Intelligence Web based tool that uses the client's high end predictive modeling technology to connect the right customer to the right restaurant with the right discounts. The client focused on ways in which advanced analytics can help both the consumers and restaurant owners, make adequate decisions leading to consumer satisfaction and enhanced profitability for restaurants. Using SpoonByte, restaurants can directly impact demand by creating real time deals that SpoonByte routes directly to end consumers. SpoonByte intelligently routes these limited time deals to selective users by using information on their preferences and proximity.

Application Screenshot

CHALLENGES

Enosis SQA team had to overcome many complex roadblocks in order to fulfill the client's stringent quality requirements, since the application has to support Restaurant owners at one end and hundreds of thousands of consumers at the other. The key features of the QA process are:

- Running intricate web security testing which ensures proper data protection and checks if web based application requirements are met when exposed to malicious input
- Focusing on testing iteratively, as often as stable code base is available and until quality is achieved from end customer's perspective
- Conducting a wide variety of tests to check the functionality of the application and to rectify the bottlenecks
- Executing intensive load testing to check the hardware sufficiency of the web server, its ability to handle increasing number of users and its capability to survive a massive spike in user traffic
- Ensuring the web application would run properly on major browsers

ENOSIS SOLUTIONS' APPROACH

The Enosis approach comprises of vigorous testing activities that are structured to maximize accuracy and optimize the time consumed so that the client can reap benefits by taking necessary measures prior of final release. Enosis' Web Application testing services include:

- Requirement Analysis
- Setting up a rapid communication channel inside the team regarding feedbacks
- Functional Testing (Sapient and Automated approach)
- Security testing
- Data-driven Testing
- Integration Testing
- System Testing
- Usability testing
- User Interface testing
- Compatibility and Multi Language Support
- Cross Browser Testing
- Exploratory and Smoke Testing
- W3C Compliance and Regression Testing
- Performance testing
- Load/Stress Testing
- User Acceptance Testing

The key testing activities that form the backbone of the Enosis' QA process are mentioned below:

- Analyze UCD workflows, identify risk areas, build scenarios and post questions to validate the requirements
- Explore other web applications and compare the features with SpoonByte UI prototypes
- Analyze database design, structures and post suggestions to build an optimized design
- Verify database objects naming pattern against database naming convention rules and report any incongruence
- Verify Nhibernate objects mapping file against class diagram and post the problems
- Build automated Unit and Integration Tests on Domain Repository classes and Controller classes
- Verify Save/Update/Find operations output and match against corresponding database storage
- Verify JSON output data and map against corresponding Mobile UI to identify the redundant or missing data field
- Generate and Test Data using appropriate tools
- Prepare Test Document Suite (Functional Test Scripts, Requirement Traceability Matrix, Summarized checklists for login and Registration module) for each Subsystem
- Simulated scenarios are executed for validating application functionalities , user interface controls and user interactions between the interfaces of the integrated modules
- Test both the Web based and Mobile based application in terms of functionality, usability, security, compatibility and performance
- Build Automated Functional Regression test suite
- Review production code to identify logical errors and to ensure proper error handling mechanism
- Exercise every independent execution path through the component to ensure all statements in the program are executed at least once
- Test all conditional statements for both true and false cases
- Organize Release Procedure (Automated build process, Semi automated Regression Test Suite execution, Release Note)
- Measure code blocks performance
- Test web server load and configuration issues

THE SOLUTION

The Enosis QA Team performed a verified set of actions for Web application testing that ensure the ability to track down all the potential bugs and errors in an application and contribute to the overall improvement of the quality of the Web Application.

THE SOLUTION (CONT.)

Functional Testing

- **Links verification:**
  - Testing all internal links
  - Testing the outgoing links from all the pages from a specific domain
  - Testing links used to send email to the administrator or other users from web pages
  - Checking if there are any orphan pages
  - Checking for broken links in all the above-mentioned links

- **Forms validation:** Forms are used to obtain information from users and to interact with them. Form validation includes checking the following in both manual and automated ways:
  - Checking all validations on each field
  - Checking for the default values of fields
  - Testing for wrong input restriction
  - Checking options to create, delete, view or modify the forms

- **Cookie testing:** Cookies are small files stored on user end machines. These are used to save the session credentials, especially those of login sessions. Cookie testing involves:
  - Testing the application by enabling or disabling the cookies from the browser options
  - Testing if the cookies are encrypted before writing to user machine
  - Testing the session cookies (i.e. cookies that expire after the session ends) for login sessions and user statistics after session ends
  - Checking the effect of cookie deletion on application security

- **HTML/CSS validation:** To optimize the site for Search Engines, HTML/CSS validation is very important. The site is validated for HTML syntax errors and checked to ensure it is friendly for search engine web crawlers. The site is checked for scripting (e.g. JavaScript) errors

- **Database testing:** Data consistency is very important in web application. Data integrity and errors that occur while users edit, delete, add forms or do any database related functionality are checked thoroughly. Database query execution, data retrieval and update are also checked. Then the retrieved results are cross checked by running the queries in database and verifying the results

Usability and UI Testing

- **Navigation Checking:**
  - Ease of navigation
  - Presence of appropriate instructions
  - Consistency of main menu on each page

- **Content Checking:**
  - Simplicity of the layout
  - Spelling mistakes
  - Presence of unnecessary links/button/input fields/form elements etc.
  - Clarity of contents on each page
  - Presence and content of sitemap
  - Images for descriptive alt information (i.e. value in "ALT" tag)
  - Color combination/pattern, size and position of image etc.

- **Connectivity Testing:** The major interfaces are Load Balancer - Application Server interface and Application Server - Database Server interface. Connectivity testing checks:
  - If all the interactions between these servers are executed properly and errors are resolved. If database or web server returns any error message for any query by application server then application server must display these error messages to the user
  - Cases such as users interruption to any transaction and web server connection reset

- **Compatibility Testing:**
  - **Browser compatibility testing:** Web applications are very dependent on the browsers. As different browsers have different configurations and settings, the compatibility of the web application must be checked to ensure that web coding is cross-browser compatible. As the web application is using Javascript or AJAX calls for UI functionality and for performing security checks or validations, more stress is given on browser compatibility testing. The web application is tested on different browsers like IE, Firefox, Chrome, Safari, Opera browsers with different versions. The web application is further tested for compatibility with browsers available on cell phones

THE SOLUTION (CONT.)

- **OS compatibility testing:** The web application has to be compatible with different platforms such as Windows, Unix, Mac, Linux etc. New technologies used in web development like graphics design, interface calls etc. may not be available in all operating systems. Therefore it is obligatory to test and ensure that the developed solution performs seamlessly across various platforms.

- **Testing Printing options:** Fonts, page alignment, page size compatibility, page graphics etc. must be validated to facilitate proper printing

Security Testing

Testing the Web application security is a vital segment of the testing process. Security testing involves:

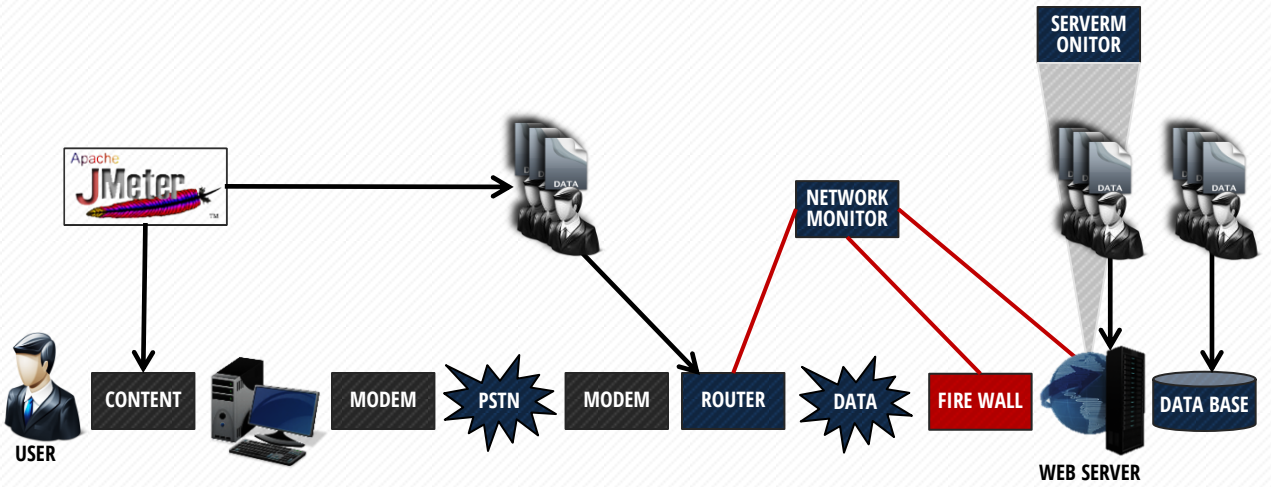
- **Approaching the application from every perspective:** A reconnaissance is performed on the Web application to see how Google presents the site and how it behaves with hacking tools such as Found Stone's Site Digger. Next, a Web vulnerability scanner such as HP's WebInspect and Acunetix Web Vulnerability Scanner is run on the application. Where possible the scans are run as both an unauthenticated and un-trusted outsider and as an authenticated and trusted user (via basic HTTP, NTLM or form authentication)

- **Test for underlying weaknesses:** Tools such as Nessus and Qualys Guard are used to identify problems such as missing patches and configurations errors in the underlying operating system and other installed applications (including the Web server itself). Back-end databases and related network infrastructure systems are also checked for weaknesses

- **Manually check for weaknesses:** Due to the scarcity of automated tools manual testing is performed on login mechanisms, form input validation and sensitive information buried in HTML and server directories. This removes risks from malicious users who might be inclined to attempt SQL and Script injection attacks

- **Testing source code:** Some tools such as DevInspect and Checkmarx are used to check for software flaws at the source. Furthermore the following precautionary measures are executed:
  - Log files of all server requests/transactions, error messages and security breach attempts are maintained on web server
  - Markup (HTML) and CSS are validated so that W3C standards are satisfied. It is essential to defend vulnerable Cross Site Scripting attacks
  - Server directory listing is disabled
  - It is verified whether the system is showing expected result in case of timeout limits
  - Some decoder is applied to automatically decode CAPTCHA image
  - Database tempering from an outside environment is restricted

Load Testing Diagram



THE SOLUTION (CONT.)

Load/Stress Testing

In order to determine the robustness of the software by testing beyond the limits of normal operations, stress tests are performed to evaluate the functionalities and error handling capabilities for large number of users.

- Testing application performance on various internet connection speeds
- **Web load testing:** It is essential for the system to sustain itself during peak hours when thousands of users log in simultaneously. Therefore the site is tested with numerous simultaneous user requests, large input data from users, simultaneous connections to database, heavy load on specific pages etc. Moreover it is necessary to check if:
  - Maximum number of connections is getting access to the server simultaneously without a busy signal
  - The returned content is verified when full load is applied to the server
  - Achieve zero downtime even when the application is running for long periods of time
  - Web page load time is measured
- **Stress testing:** Stress is generally exerted on input fields, login and sign up areas to check how the system responds and recovers from crashes

TOOLS AND TECHNOLOGIES

Programming Language: C#.
Frameworks: Asp.NET MVC, WCF, Windows Service
Web Technologies: HTML, XHTML, JavaScript, CSS, AJAX
Relational Database Management System: PostgreSQL, Microsoft SQL Server
Object Relational Mapping (ORM) Tool: NHibernate
Automated Functional Testing Tool: Selenium, WATIN
XHTML & CSS Validator Tool: Total Validator, W3C Validator, W3C mobileOK Checker, CSS Validator
Broken Link Checker Tool: Xenu's Link Sleuth, W3C Link Checker
Spell Checker Tools: Total Validator, IESPELL
Cookie Watcher/Viewer Tools: IECookiesView v1.74, MozillaCookiesView v1.27, Safari Cookies, Fire Cookies
Webpage Control Inspectors: Firebug (Firefox), DebugBar (IE)
Webpage Performance Analyzers: YSlow (Firebug extension & Add-on)
HTTP Packet Analyzer/Sniffers: WebBug, WireShark, Live HTTP Header
Load/Stress/Performance Testing Tool: Jmeter
SQL Injection: Hackbar, SQL Injection!
CAPTCHA Decoder: PWNtcha
User-scripts Executor: GreaseMonkey
Maximum Length Input Generator: PerlClip
Test Data Generator: Advance Data Generator