

Testing Your Web Application

A Quick 10-Step Guide

by Krishen Kota, PMP

Interested in a quick checklist for testing a web application? The following 10 steps cover the most critical items that I have found important in making sure a web application is ready to be deployed. Depending on size, complexity, and corporate policies, modify the following steps to meet your specific testing needs.

Step 1 - Objectives

Make sure to establish your testing objectives up front and make sure they are measurable. It will make your life a lot easier by having written objectives that your whole team can understand and rally around. In addition to documenting your objectives, make sure your objectives are prioritized. Ask yourself questions like "What is most important: minimal defects or time-to-market?"

Here are two examples of how to determine priorities:

If you are building a medical web application that will assist in diagnosing illnesses, and someone could potentially die based on how correctly the application functions, you may want to make testing the correctness of the business functionality a higher priority than testing for navigational consistency throughout the application.

If you are testing an application that will be used to solicit external funding, you may want to put testing the aspects of the application that impact the visual appeal as the highest testing priority.

Your web application doesn't have to be perfect; it just needs to meet your intended customer's requirements and expectations.

Step 2 – Process and Reporting

Make sure that everyone on your testing team knows his or her role. Who should report what to whom and when? In other words, define your testing process. Use the following questions to help you get started:

- How will issues be reported?
- Who can assign issues?
- How will issues be categorized?
- Who needs what report and when do they need it?
- Are team meetings scheduled in advance or scheduled as needed?

You may define your testing process and reporting requirements formally or informally, depending on your particular needs. The main point to keep in mind is to organize your team in a way that supports your testing objectives and takes into account the individual personalities on your team. One size never fits all when dealing with people.

Step 3 - Tracking Results

Once you start executing your test plans, you will probably generate a large number of bugs, issues, defects, etc. You will want a way to easily store, organize, and distribute this information to the appropriate technical team members. You will also need a way to keep management informed on the status of your testing efforts. If your company already has a system in place to track this type of information, don't try to reinvent the wheel. Take advantage of what's already in place.

If your company doesn't already have something in place, spend a little time investigating some of the easy-to-setup online systems such as the one found at <http://www.adminitrack.com>. By using an online system, you can make it much easier on yourself by eliminating the need to install and maintain an off-the-shelf package.

Step 4 - Test Environment

Set up a test environment that is separate from your development and production environment. This includes a separate web server, database server, and application server if applicable. You may or may not be able to utilize existing computers to setup a separate test environment.

Create an explicitly defined procedure for moving code to and from your test environment and make sure the procedure is followed. Also, work with your development team to make sure each new version of source code to be tested is uniquely identified.

Step 5 - Usability Testing

In usability testing, you'll be looking at aspects of your web application that affect the user's experience, such as:

- How easy is it to navigate through your web application?
- Is it obvious to the user which actions are available to him or her?
- Is the look-and-feel of your web application consistent from page to page, including font sizes and colors?

The book, "Don't Make Me Think! A Common Sense Approach to Web Usability" by Steve Krug and Roger Black, provides a practical approach to the topic of usability. I refer to it often, and recommend it highly.

In addition to the traditional navigation and look-and-feel issues, Section 508 compliance is another area of importance. The 1998 Amendment to Section 508 of the Rehabilitation Act spells out accessibility requirements for individuals with certain disabilities.

For instance, if a user forgets to fill in a required field, you might think it is a good idea to present the user with a friendly error message and change the color of the field label to red or some other conspicuous color. However, changing the color of the field label would not really help a user who has difficulty deciphering colors. The use of color may help most users, but you would want to use an additional visual clue, such as placing an asterisk beside the field in question or additionally making the text bold.

For more details, refer to <http://www.section508.gov>. Another great resource that can help analyze your HTML pages for Section 508 compliance can be found at <http://www.cast.org/bobby/>. If you are working with the United States federal government, Section 508 compliance is not only good design; it most likely is a legal requirement. You may want to utilize the following information regarding techniques for accessibility evaluation and repair tools, which can be found at <http://www.w3.org/TR/AERT>.

Step 6 – Unit Testing

Unit testing is focused on verifying small portions of functionality. For example, an individual unit test case might focus on verifying that the correct data has been saved to the database when the Submit button on a particular page is clicked.

An important subset of unit testing that is often overlooked is range checking. That is, making sure all the fields that collect information from the user, can gracefully handle any value that is entered. Most people think of range checking as making sure that a numeric field only accepts numbers. In addition to traditional range checking make sure you also check for less common, but just as problematic exceptions. For example, what happens when a user enters his or her last name and the last name contains an apostrophe, such as O'Brien? Different combinations of databases and database drivers handle the apostrophe differently, sometimes with unexpected results. Proper unit testing will help rid your web application of obvious errors that your users should never have to encounter.

Step 7 - Verifying the HTML

Hyper Text Markup Language (HTML) is the computer language sent from your web server to the web browser on your users' computer to display the pages that make up your web application. The World Wide Web Consortium (www.w3.org) manages the HTML specification. One major objective of HTML is to provide the ability for anyone from anywhere to access information on the World Wide Web. This concept generally holds true if you conform strictly to the relevant version of the HTML specification that you will support. Unfortunately, in the real world, it is possible for a developer to inadvertently use a proprietary HTML tag that may not work for all of your intended users.

Verifying HTML is simple in concept but can be very time consuming in practice. A good place to start is with the World Wide Web Consortium's free HTML Validation Service (<http://validator.w3.org>). There are also other online and downloadable applications to help in this area such as Net Mechanic (<http://www.netmechanic.com>). There are two main aspects of verifying the validity of your HTML. First, you want to make sure that

your syntax is correct, such as verifying that all opening and closing tags match, etc. Secondly, you want to verify how your pages look in different browsers, at different screen resolutions, and on different operating systems. Create a profile of your target audience and make some decisions on what browsers you will support, on which operating systems, and at what screen resolutions.

In general, the later versions of Microsoft Internet Explorer are very forgiving. If your development team has only been using Internet Explorer 6 on high-resolution monitors, you may be unpleasantly surprised when you see your web application on a typical user's computer. The sooner you start verifying your HTML, the better off your web application will be.

Step 8 - Load Testing

In performing load testing, you want to simulate how users will use your web application in the real world. The earlier you perform load testing the better. Simple design changes can often make a significant impact on the performance and scalability of your web application. A good overview of how to perform load testing can be found on Microsoft's Developer Network (MSDN) website.

<http://msdn.microsoft.com/msdnmag/issues/03/01/LoadTesting/default.aspx>

A topic closely related to load testing is performance tuning. Performance tuning should be tightly integrated with the design of your application. If you are using Microsoft technology, the following article is a great resource for understanding the specifics of tuning a web application.

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vsent7/html/vxconPerformance.asp>

People hate to wait for a web page to load. As general rule, try to make sure that all of your pages load in 15 seconds or less. This rule will of course depend on your particular application and the expectations of the people using it.

Step 9 - User Acceptance Testing

By performing user acceptance testing, you are making sure your web application fits the use for which it was intended. Simply stated, you are making sure your web application makes things easier for the user and not harder. One effective way to handle user acceptance testing is by setting up a beta test for your web application.

One article to help you get started planning an effective beta test is: Supercharged Beta Test by Joshua Grossnickle and Oliver Raskin, May 14, 2001 which can be found at: <http://hotwired.lycos.com/webmonkey/01/20/index1a.html?tw=design>. This article points out the critical aspects of setting up a beta test including how to identify beta testers and how to obtain their feedback. The main point to remember in user acceptance testing is to listen to what the people using your web application are saying. Their feedback will be critical to the ultimate success of your web application.

Step 10 - Testing Security

With the large number of highly skilled hackers in the world, security should be a huge concern for anyone building a web application. You need to test how secure your web application is from both external and internal threats. The security of your web application should be planned for and verified by qualified security specialists.

If you think security is a subject that is over-hyped, check out Steve Gibson's account of how a 13 year old hacker took his company's website down for an extended period of time at will. You can find this eye-opening security case study at:

<http://grc.com/dos/grcdos.htm>

Some additional online resources to help you stay up to date on the latest Internet security issues include:

CERT Coordination Center

<http://www.cert.org/>

Computer Security Resource Center

<http://csrc.nist.gov/>

After performing your initial security testing, make sure to also perform ongoing security audits to ensure your web application remains secure over time as people and technology change.

Testing a web application can be a totally overwhelming task. The best advice I can give you is to keep prioritizing and focusing on the most important aspects of your application and don't forget to solicit help from your fellow team members.

By following the steps above coupled with your own expertise and knowledge, you will have a web application you can be proud of and that your users will love. You will also be giving your company the opportunity to deploy a web application that could become a run away success and possibly makes tons of money, saves millions of lives, or slashes customer support costs in half. Even better, because of your awesome web application, you may get profiled on CNN, which causes the killer job offers to start flooding in.

Proper testing is an integral part of creating a positive user experience, which can translate into the ultimate success of your web application. Even if your web application doesn't get featured on CNN, CNBC, or Fox News, you can take great satisfaction in knowing how you and your team's diligent testing efforts made all the difference in your successful deployment.

Copyright © 2005 Krishen Kota, All Rights Reserved

Krishen Kota is a 15+year veteran of the information technology consulting industry and is a Certified Project Management Professional. Krishen serves as President of AdminiTrack, Inc. (www.adminitrack.com), which provides a web-based issue and defect tracking application designed specifically for professional software development teams. Krishen can be contacted via email at kkota@adminitrack.com.