

# Qualität von Webpräsenzen - Eine Herausforderung

(Originaltitel: The WebSite Quality Challenge)

**Edward Miller**

Software Research, Inc.  
1663 Mission Street, Suite 400  
San Francisco, CA 94103 USA

übersetzt von  
Dr. Jürgen Pitschke  
Business Consulting Saxony  
PF 160 116  
01287 Dresden, Germany

© Copyright 2004 by Software Research, Inc.

## ZUSAMMENFASSUNG

Eine Webpräsenz ist potentiell sofort einem weltweiten Publikum sichtbar. Qualität und Zuverlässigkeit sind daher entscheidend. Die besondere Natur des WWW und von Webseiten bringen besondere Herausforderungen mit sich. Webmaster, Anwendungsentwickler von Internetanwendungen und QA-Manager von Webseiten benötigen Werkzeuge und Methoden, die den neuen Anforderungen gerecht werden. Der instrumentierte Test mittels spezieller Web-Testsoftware hat das Potential diese Herausforderungen zu bestehen.

## EINFÜHRUNG

Webpräsenzen sind etwas neues in der Welt der Softwarequalität! Innerhalb von Minuten nach dem „Life“-Schalten kann eine Webanwendung mehrere tausend Benutzer haben, viel mehr als eine konventionelle, nicht web-basierte Anwendung. Das Unmittelbare des Internet erzeugt eine direkte Erwartung von Qualität und schneller Bereitstellung der Anwendung. Zugleich wird durch die technische Komplexität von Webpräsenzen und die verschiedenen Browservarianten das Testen und die Qualitätskontrolle schwieriger und in gewisser Weise subtiler. Das automatisierte Testen von Webseiten ist Chance und Herausforderung zugleich.

## QUALITÄT UND VERFÜGBARKEIT VON WEBSEITEN – DEFINIERT

Eine Webseite ist wie jede andere Software: es gilt nicht nur ein alles beinhaltendes Qualitätsmaß und auch mehrere Metriken mögen nicht zutreffen. Die Überprüfung der benutzerkritischen Eindrücke der „Qualität“ und „Verfügbarkeit“ erhalten zudem ein neues Gewicht.

**Qualitätsdimensionen.** Qualität hat viele Dimensionen und jede Einheit betrifft eine bestimmte Präsenz in unterschiedlichem Maß. Hier sind einige Kriterien:

- *Zeit:* Web-Seiten ändern sich oft und schnell? Wie stark hat sich eine Webseite seit dem letzten Upgrade geändert? Wie können die geänderten Teile hervorgehoben werden?
- *Struktur:* Wie gut passen alle Bestandteile der Webpräsenz zusammen? Funktionieren alle Links innerhalb und außerhalb der Seite? Werden alle Grafiken angezeigt? Gibt es Teile der Präsenz, die nicht verbunden sind?
- *Inhalt:* Stimmt der Inhalt kritischer Seiten? Existieren Schlüsselphrasen durchgängig in änderbaren Seiten? Bleibt die Qualität des Inhalts kritischer Seiten von Version zu Version erhalten? Wie steht es mit dynamisch generierten HTML-Seiten?

- *Genauigkeit und Konsistenz:* Sind die heute heruntergeladenen Seiten dieselben wie gestern? Zumindest nah genug? Sind die dargestellten Daten genau genug? Wie kann das beurteilt werden?
- *Antwortzeiten und Latenz:* Antwortet der Webserver auf Browser-Anforderungen innerhalb bestimmter Parameter? In einem eCommerce-Kontext: Wie lang ist die End-zu-End-Antwortzeit nach einem SUBMIT? Gibt es Teile der Seite, die so langsam sind, dass ein Anwender sie nicht mehr nutzen wird?
- *Laufzeitverhalten:* Ist die Browser-Web-Webpräsenz-Web-Browser-Verbindung schnell genug? Wie verändert sich das Laufzeitverhalten über den Tag, unter Last und Benutzung? Reicht das Laufzeitverhalten für eCommerce-Anwendungen aus? Eine Antwortzeit von 10 Minuten ist für eCommerce sicher nicht akzeptabel!

**Auswirkungen der Qualität.** Qualität entsteht im Kopf des Anwenders. Eine Webpräsenz mit schlechter Qualität, mit vielen fehlerhaften Seiten und fehlenden Bildern, mit CGI-Fehlermeldungen etc. wirkt sich möglicherweise negativ auf die Kundenbeziehung aus, generiert ein schlechtes Firmenimage und resultiert letztendlich in verlorenem Umsatz. Sehr komplexe Webpräsenzen überfordern manchmal den Anwender.

Die Kombination von Komplexität einer Webseite und schlechter Qualität kann das Aus für eine eCommerce-Anwendung bedeuten. Unzufriedene Benutzer werden schnell zu einer anderen Seite wechseln, ohne einen einzigen guten Eindruck zu behalten!

## ARCHITEKTUR VON WEBPRÄSENZEN

Webpräsenzen sind komplex und diese Komplexität (welche die Präsenz selbstverständlich zugleich von anderen unterscheidet und heraushebt) ist ein Hindernis bei der Sicherung der Qualität der Webseite. Nimmt man noch mehrere Autoren, schnelle Updates und Änderungen hinzu, wird die Sache noch komplizierter.

Betrachten wir die wichtigsten Bestandteile einer Webpräsenz aus Qualitätssicht.

**Browser.** Die Webpräsenz wird durch einen Browser betrachtet. Es gibt viele verschiedene Browser und Browser-Optionen. Eine gut gemachte Webseite wird so entworfen, dass sie mit möglichst vielen Browsern gut aussieht. Das bringt einen de-facto-Standard hervor: Webseiten dürfen nur Konstrukte benutzen, die mit der Mehrheit der Browser funktionieren. Aber auch das lässt noch genügend Raum für Kreativität und eine Menge von technischen Schwierigkeiten.

**Anzeige-Technologien.** Was im Browser angezeigt wird, entsteht aus verschiedenen Quellen:

- *HTML.* Es werden verschiedene HTML-Versionen unterstützt und die Webseite sollte besser mit einer kompatiblen HTML-Version gebaut sein. Und das sollte auch überprüfbar sein.
- *Java, JavaScript, ActiveX.* Offensichtlich sind JavaScript und Java-Applets Teil jeder ernsthaften Webseite. Daher muss der Qualitätsprozess diese auch unterstützen. Auf der Windows-Plattform müssen ActiveX-Controls ebenfalls beherrscht werden.
- *CGI-Bin Skripte.* CGI-Skripte sind mit Benutzeraktionen verbunden (typischerweise aus einer FORM-Passage oder direkt aus dem HTML oder möglicherweise auch aus einem Java-Applet heraus). Die verschiedenen Typen von Skripten (Perl, awk, Shell-Skripte, etc.) müssen beherrscht werden und Tests müssen „End-zu-End“-Operationen prüfen. Das ist insbesondere für eCommerce-Situationen entscheidend.
- *Datenbank-Zugriff.* In eCommerce-Anwendungen werden Daten aus Datenbanken abgerufen oder in Datenbanken abgelegt. Wie verhält sich diese Interaktion bei wirklicher Benutzung? Entspricht das Ergebnis den Erwartungen, wenn „korrekte“ oder „spezifizierte“ Eingaben getätigt werden?

Manche Zugriffe zu Informationen der Datenbank können abhängig von der Anwendung zulässig oder unzulässig sein, aber das wird typischerweise mit anderen Mitteln geprüft.

**Navigation.** Benutzer gehen zu bestimmten Seiten, klicken auf Links und Bilder, etc. Die Navigation in einer Webpräsenz ist vielfältig und muss schnell und fehlerfrei erfolgen.

**Objektmodus.** Die Anzeige ändert sich dynamisch, die einzigen Konstanten sind die „Objekte“, die für die Anzeige benutzt werden. Das sind keine Objekte im Sinne der objektorientierten Programmierung, aber sie müssen in genau diesem Sinne betrachtet werden. Das Testwerkzeug

muss in der Lage sein URL-Links, Formulare, Tabellen, Anker, Knöpfe aller Typen in einer „Objekt“-Art zu behandeln, damit die Validierung unabhängig von der Präsentation ist.

**Server Antwort.** Wie schnell der Hostserver der Webpräsenz reagiert, beeinflusst ob ein Benutzer weitergeht. Auch die Internet-Netzlant hat offensichtlich einen Einfluss, aber das ist ein Faktor, der oft außerhalb der Kontrolle des Webmasters liegt, zumindest unter dem Aspekt wie eine Webpräsenz gestaltet ist. Im Gegensatz scheint das mehr eine Frage der Serverhardwareleistung und des Durchsatzes zu sein. Aber, wenn eine Webpräsenz sehr populär wird – und das kann faktisch über Nacht passieren – werden Lastverteilung und Tuning zu echten Fragen, die dem Webmaster auferlegt werden.

**Interaktion & Feedback.** Für passive, nur Inhalt präsentierende Seiten ist die Verfügbarkeit die einzige Frage. Für Webpräsenzen die dem Nutzer Interaktionsmöglichkeiten bieten, ist die Verfügbarkeit und Geschwindigkeit der Interaktionen ein wichtiger Aspekt.

**Gleichzeitige Benutzung.** Können mehrere Benutzer mit der Webseite interagieren? Können Sie sich „im Wege stehen“? Webpräsenzen gleichen häufig konventionellen Client/Server-Strukturen. Aber mit mehreren Nutzern in verschiedenen Plätzen kann eine Webanwendung wesentlich komplizierter werden als eine komplexe Anwendung.

## AUTOMATISCHE QUALITÄTSSICHERUNG VON WEBSEITEN UND ANWENDUNGEN

Die Sicherung der Qualität einer Webpräsenz oder einer Webanwendung erfordert die Durchführung einer Menge von Test, automatisiert und wiederholbar, welche die geforderten Eigenschaften und Verhalten zeigen. Werkzeuge, die den Test unterstützen sollen, müssen folgende Elemente erfüllen:

**Testsitzungen.** Typische Testelement beinhalten die folgenden Charakteristika:

- *Browser Unabhängig.* Tests sollen realistisch sein, aber nicht von einem bestimmten Browser abhängen, dessen Charakteristiken Probleme der Webseite verbergen können.
- *Kein Buffering und Caching.* Lokales Caching und Buffering – oft ein Weg, um die Performance zu steigern – sollte ausgeschaltet sein, so dass die Zeitmessungen ein wahres Bild von der Browser-Web-Webpräsenz-Web-Browser-Antwortzeit wiedergeben.
- *Fonts und Einstellungen.* Die meisten Browser unterstützen viele Fonts und Voreinstellungen für die Anzeige. Diese Einstellungen dürfen die Beurteilung und Sicherung der Qualität der Webpräsenz nicht beeinflussen.
- *Objektmodus.* Eingabe-Felder, Push Button, Radio Button, Check Boxen, etc. Alle diese Konstrukte sollten im Objektmodus, d.h. unabhängig von Fonts und Einstellungen, behandelbar sein.

Operationen im Objektmodus sind wichtig, um das Investment in Tests zu schützen und die fortdauernde Funktion der Tests zu sichern, wenn eine Webpräsenz sich ändert. Wenn ein Knopf oder Formular einen anderen Platz auf der Seite bekommt (was häufig geschieht), soll der Test immer noch funktionieren.

Wenn ein Knopf oder ein anderes Objekt gelöscht wird, sollte das natürlich erkannt werden! Das Hinzufügen von Objekten zu einer Seite impliziert ganz klar ein Überarbeiten des Tests.

- *Tabellen und Formulare.* Selbst wenn das Layout einer Tabelle oder eines Formulars im Browser variiert, soll der Test unabhängig davon weiter funktionieren.
- *Frames.* Fenster mit mehreren Frames sollten einfach verarbeitet werden, z.B. als wenn sie mehrere Single-Page-Frames wären.

**Testkontext.** Tests müssen aus einem Browserniveau ausgeführt werden. Dafür gibt es zwei entscheidende Gründe: (1) Der Browser ist das Mittel mit der ein Anwender die Webpräsenz sieht. Daher sind Tests auf dem Browserlevel am realistischsten. (2) Browser-basierte Tests können sowohl lokal als auch über das Web ausgeführt werden. Die lokale Ausführung ist besonders für die

Qualitätskontrolle geeignet, aber nicht für Laufzeitmessungen, wo die Antwortzeit inklusive Webvariabler Verzögerungen mit realem Bezug essentiell ist.

## DER PROZESS DER WEBSEITEN-VALIDIERUNG

Die Bestätigung der ausgeführten Tests ist der Schlüssel zur Sicherung der Qualität von Webpräsenzen – und ist die größte Herausforderung zugleich. Im folgenden sind vier Schlüsselbereiche benannt, in denen die Testautomatisierung eine signifikante Auswirkung hat.

**Operativer Test.** Einzelne Testschritte enthalten eine Vielzahl von Prüfungen individueller Seiten der Webpräsenz.

- *Seitenqualität.* Ist die gesamte Seite identisch mit einer früheren Version? Sind die wichtigen Bereiche des Texts noch gleich oder verschieden?
- *Tabellen-, Formular-Qualität.* Sind alle Teile einer Tabelle oder eines Formulars verfügbar? Ist das Layout korrekt? Ist ausgewählter Text „an der richtigen Stelle“?
- *Seiten-Beziehungen.* Sind alle Links auf einer Seite noch dieselben wie zuvor? Gibt es neue oder fehlende Links?
- *Performance, Antwortzeiten.* Ist die Antwortzeit für eine Benutzeraktion dieselbe (in bestimmten Grenzen) ?

**Test Suites.** Typischerweise wird man Dutzende oder Hunderte (oder Tausende?) Tests haben und man möchte diese Tests in verschiedenen Modi ausführen:

- *Unbeobachteter Test.* Individuelle Tests oder Testgruppen sollen einzeln oder parallel von einer oder mehreren Workstation ausgeführt werden.
- *Test im Hintergrund.* Tests sollten von mehreren „im Hintergrund“ laufenden Browsern ausgeführt werden (auf einer entsprechend ausgestatteten Workstation).
- *Verteilte Tests.* Unabhängige Teile der Testsuite sollen von verschiedenen Workstation ohne Konflikte ausgeführt werden.
- *Performance Test.* Zeitmessungen in Performance-Tests sollten auf dem Millisekunden-Niveau aufgelöst werden. So ergibt sich eine starke Basis für Durchschnittsdaten.
- *Zufälliger Test.* Die Möglichkeit bestimmte Teile des Tests zufällig auszuführen sollte gegeben sein.
- *Fehler-Recovery.* Browser-Fehler wegen Benutzereingaben sind selten. Test-Suiten sollten aber die Möglichkeit der Synchronisierung nach einem Fehler trotzdem bieten.

**Validierung des Inhalts.** Abgesehen vom dynamischen Verhalten einer Webpräsenz sollte der Inhalt überprüfbar sein, entweder exakt oder in Näherung. Einige der Möglichkeiten sind:

- *Strukturell.* Alle Links und Anker stimmen mit früheren „Baseline“-Daten überein. Bilder sollten durch die Bytezahl und/oder Dateityp oder andere Dateieigenschaften charakterisierbar sein.
- *Checkpoints, Exakte Reproduzierbarkeit.* Einzelne oder mehrere oder auch alle Textelemente einer Seite sollen als „notwendig übereinstimmend“ markierbar sein.
- *Statistiken.* Seitenstatistiken (z.B. Zeilenzahl, Wörter, Bytezahl, Prüfsummen, etc.)
- *Ausgewählte Images/Fragmente.* Der Tester soll die Option haben, Sektionen eines Bildes auszuwählen und zu bestimmen, dass das ausgewählte Bild mit einer späteren Ausführung übereinstimmen muss. Das sollte für mehrere Bilder oder Bildfragmente möglich sein.

### Last-Simulation.

Eine Last-Analyse muss durch einen speziellen Browser, der wie ein menschlicher Anwender agiert, vorgenommen werden. Das sichert, dass die Performance-Messung der wahren Leistung entspricht und kein Verhalten unter simulierten, aber unrealistischen Bedingungen ist.

Sitzungen sollten live aufgezeichnet werden oder von Aufzeichnungen editiert werden, um originalgetreues Timing zu sichern. Die Geschwindigkeit sollte anpassbar sein durch Beschleunigungs- oder Verzögerungsraten und Intervalle.

Lastgenerierung soll wie folgt ausführbar sein:

- *Einzelner Browser.* Eine Sitzung mit einer oder mehreren Antworten, die von einem Browser abgespielt wird. Die Zeitdaten werden in einer Datei für die spätere Analyse gespeichert.
- *Mehrere unabhängige Browser.* Mehrere Sitzungen mit einer oder mehreren Antworten, die von mehreren Browsern abgespielt wird. Die Zeitdaten werden in einer Datei für die spätere Analyse gespeichert. Statistische Methoden sind für ein komplexes aber allgemeingültiger Performance-Modell notwendig.

- *Mehrere koordinierte Browsers.* Das ist die komplizierteste Form – zwei oder mehr Browser, deren Aktionen koordiniert ablaufen. Dafür müssen besondere Synchronisations- und Steuermöglichkeiten verfügbar sein.

## **ZUSAMMENFASSUNG**

Alle diese Anforderungen und Vorgaben setzen Randbedingungen für Testautomatisierungswerkzeuge zur Qualitätssicherung einer Webpräsenz. Gleichzeitig bieten sie einen realen Ansatz um die Möglichkeiten manueller, durch Menschen ausgeführter Tests zu verbessern. Bessere und stabilere Webpräsenzen sind das Ergebnis.

## **ANMERKUNGEN ZUR ÜBERSETZUNG VON JÜRGEN PITSCHKE:**

Die erste Version dieses White Papers ist bereits drei Jahre alt. Einige der Aussagen scheinen dem Leser möglicherweise nicht mehr relevant. So hat sich die Anzahl der verfügbaren Browser verringert, die Möglichkeiten der Browser gleichen sich immer mehr an. Das ist jedoch nur auf den ersten Blick richtig. Haben sich die Schwierigkeiten aus der Verwendung unterschiedlicher Browser im Laufe der Zeit etwas verringert, sind gleichzeitig neue Fragen entstanden. Webseiten werden heute für das Auffinden in Suchmaschinen optimiert, zusätzlich zu den im Artikel benannten Technologien hat sich XML als Webtechnologie etabliert. Dabei gelten die von Edward Miller getroffenen Aussagen wiederum voll und ganz.

Anmerkungen und Kommentar sind gern willkommen. Bitte senden Sie diese an [jpitschke@sw-products.de](mailto:jpitschke@sw-products.de).