



UNIVERSITY OF OREGON
APPLIED INFORMATION MANAGEMENT

Presented to the Interdisciplinary
Studies Program:
Applied Information Management
and the Graduate School of the
University of Oregon
in partial fulfillment of the
requirement for the degree of
Master of Science

Benefits of Using Automated Software Testing Tools to Achieve Software Quality Assurance

CAPSTONE REPORT

**Michael T. Grater
Intermediate Methods Analyst
United Parcel Service**

University of Oregon
Applied Information
Management
Program

June 2005

722 SW Second Avenue
Suite 230
Portland, OR 97204
(800) 824-2714

[This page intentionally left blank.]

Approved by

Dr. Linda F. Ettinger
Academic Director, AIM Program

[This page intentionally left blank.]

Abstract for:
Benefits of Using Automated Software Testing Tools
to Achieve Software Quality Assurance

The increasing complexity of software development demands that developers use automated software testing tools. This study analyzes selected literature published between 1997 and 2005 to provide an analysis of benefits of automated software testing tools to increase software quality assurance. Benefits are presented for software engineers and business analysts and classified in relation to five quality factors (in reference to web applications) described by QA Labs, including quality, reliability, recoverability, security, usability and performance.

[This page intentionally left blank.]

Table of Contents

CHAPTER I – PURPOSE OF STUDY	1
Brief Purpose	1
Full Purpose	4
Significance of Study	10
Limitations	12
Definitions.....	15
Problem Area	20
CHAPTER II – REVIEW OF REFERENCES	23
References Related to Software Testing Strategies	24
References Related to Software Quality Assurance	25
References Related to Automated Testing Software	26
References Related to Web Based Applications.....	35
CHAPTER III – METHOD	39
Data Collection	40
Data Analysis	42
Data Presentation	46
CHAPTER IV – ANALYSIS OF DATA	49
CHAPTER V – CONCLUSION	53
Bibliography	59
Appendix A - Sources Used in Content Analysis.....	63
Appendix B - Contextual Discussions Related to Software Quality Assurance, Automated Testing Software and Web Based Applications.....	65

[This page intentionally left blank.]

List of Tables

Table 1: Testing Tools (from Dustin, 2003, p. 162)	21
Table 2: Template: Benefits Of Utilizing An Automated Testing Tool	46
Table 3: List of Coding Phrases and Associated Terms	50
Table 4: Full List of Coding Terms.....	50
Table 5: Benefits Of Using Software Automated Tools.....	54
Table 6: List of common themes in the data analysis results.....	56

[This page intentionally left blank.]

List of Figures

Figure 1: The Role of the Analyst.....8
Figure 2: Three Types of Requirements9

[This page intentionally left blank.]

CHAPTER I – PURPOSE OF STUDY

Brief Purpose

The purpose of this study is to examine the value of using automated software testing tools to increase software quality assurance for Internet-based applications (Mayer, 1998; Pepe, 2000). According to Schwartz (2003), the increasing complexity of software development demands that developers use automated software testing tools.

Software testing is a costly and time-consuming activity in the software development life cycle (Rankin, 2002). Testing is also an integral part of the development process (Mayer, 1998, p. 1). Testing and software quality assurance play critical roles in the successful implementation of any application (Mayer, J, 1998).

Software test automation refers to the activities and efforts that intend to automate engineering tasks and operations in a software test process using well-defined strategies and systematic solution (Gao & Tsao & Wu, 2003). Automation testing software is software that tests application performance and functionality, (Hayes, Whiting, 2002) and aids developers in debugging and releasing applications faster (Pepe, 2000, p. 57). Another term for this is automated testing tools (software tools that automate the testing process in order to reduce effort, timescales, and cost in the testing of software (Watkins, 2001).

The research method used for this study is literature review (Leedy & Ormrod, 2001). Literature review is chosen as an appropriate method for this study because the majority of data on the subject exist in documentation

published in technical and academic journals, books, and as a result of studies performed by professional societies. Literature is collected from materials published between December 1997 and April 2005.

Once the literature is selected, these resources are subjected to content analysis. A content analysis is a detailed and systematic examination of the contents of a particular body of material for the purpose of identifying patterns, themes, or biases (Leedy and Ormrod, 2001). The form of content analysis selected for this paper is a conceptual analysis (Palmquist et al., 2005). The goal of the conceptual analysis is to identify the occurrence of the specific terms "Software Quality Assurance", "Automated Testing Software", "Software Testing Strategies" and "Web Based Applications" along with the context of the discussion of each of these. Additional similar terms that emerge are noted, along with the relevant context.

The results of the conceptual analysis are presented in the form of a large table (see Appendix B – Contextual Discussions Related to Software Quality Assurance, Automated Testing Software and Web Based Applications) that presents contextual discussions, excerpted from the selected literature, related to the initial set of coded terms and any additional terms that emerge. Then, the results are framed for business analysts and presented in the form of a final outcome table (see Table 5: Benefits Of Using Software Automated Tools). Table 5 is designed to categorize the results of the conceptual analysis, using the initial set of terms listed above, i.e., "Software Quality Assurance", "Automated Testing Software", "Software Testing

Strategies" and "Web Based Applications" in a way that reveals the meaning of the context of the discussions in relation to potential benefits reported, as a result of using automated testing software. Benefits are classified in terms of five key factors of quality in reference to Web Applications, as described by QA Labs. These factors are: quality, reliability, recoverability, security, usability and performance. The hope of the researcher is that this process will more clearly demonstrate ways in which software testing strategies and the use of automated testing software can increase software quality assurance in web-based applications.

Although the outcome of this study is intended to be useful to project leaders, senior programmers, and quality assurance analysts, the resource provided is specifically designed to be of use to business analysts who deal with Internet-based object oriented software applications (Lethbridge, T & Laganière, R, 2001).

Full Purpose

The purpose of this study is to demonstrate that a relationship exists between software testing strategies that utilize automated software testing tools and increased software quality assurance for Internet-based applications (Mayer, 1998; Pepe, 2000). Business applications today are large, multi-tiered, distributed, and integrated (Schwartz, 2003). There is a wide range of sophistication in web applications from a simple company website "brochure ware" to sites like Yahoo or Amazon with complex search engines and order fulfillment (QA Labs, 2000, pg 1).

Software quality is defined, according to the Institute of Electrical and Electronics Engineers (IEEE), as "the degree to which software meets customer or user needs or expectations" (Gao & Tsao & Wu, 2003). According to this definition, quality assurance can simply be interpreted as assuring customer satisfaction (Gao & Tsao & Wu, 2003). According to IEEE Standard 610.12-1990, "IEEE Standard Glossary of Software Engineering Terminology" (Copeland, 2004) "testing" is defined as:

"The process of operating a system or component under specified conditions, observing or recording the results, and making an evaluation of some aspect of the system or component."

At its core, testing is the process of comparing "what is" with "what ought to be" (Copeland, 2004). Software companies rely on the testing phase of an application to isolate bugs and defects (Stahl, 2002; Wells, 2002). It is important to test an application for functionality, scalability and performance before it becomes active in production, once it is active in

production and after it is active in production (Schwartz, 2003). However, Mayer (1998) states that testing software is an activity that code developers and other staff view as an impediment to time-to-market goals (Mayer, J, 1998, p. 3).

The following paragraph explains test automation:

Over the past few years, tools that help programmers quickly create applications with graphical user interfaces have dramatically improved programmer productivity. This has increased the pressure on testers, who are often perceived as bottlenecks to the delivery of software products. Testers are being asked to test more and more code in less and less time. Test automation is one way to do this, as manual testing is time consuming. As and when different versions of a software are released, the new features will have to be tested manually time and again. But, now there are tools available that help the testers in the automation of the GUI which reduce the test time as well as the cost, other test automation tools support execution of performance tests. Many test automation tools provide record and playback features that allow users to record interactively user actions and replay it back any number of times, comparing actual results to those expected. (http://en.wikipedia.org/wiki/Test_automation)

Literature review is chosen as an appropriate method for this study because the majority of data on the subject exist in documentation published in technical journals, trade publications, and academic research and as a result of studies performed by professional societies. The following content

areas are searched for pertinent literature: "Software Quality Assurance", "Automated Testing Software", "Software Testing strategies" and "Web Based Applications". The term "Web Based Applications" refers to Internet-based object oriented software applications (Lethbridge, T & Lavalieres, R, 2001). The purpose of using the specific term is to narrow the focus of the paper to deal specifically with applications that are not mainframe based, but are used in a web-based environment (Schwinger et al., 2003). These applications have a Graphical User Interface (GUI), and are delivered to users from a web server over a network such as the World Wide Web or an intranet (http://en.wikipedia.org/wiki/Web_application).

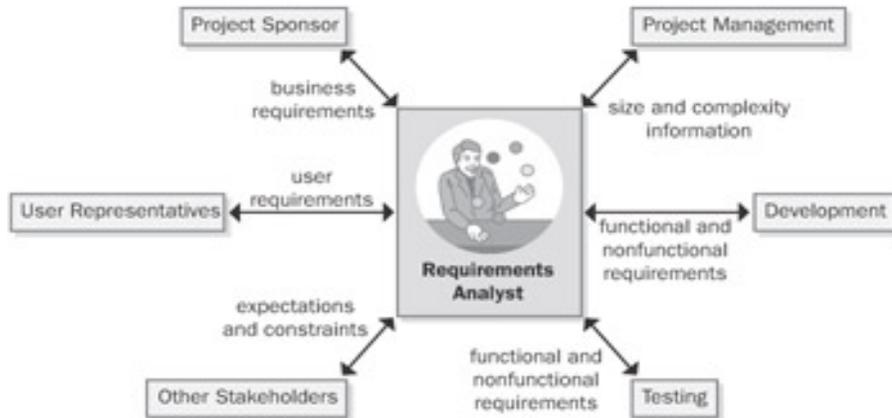
Content analysis is selected as the best strategy for data analysis. The form of content analysis selected for this paper is a conceptual analysis (Palmquist et al., 2005). In conceptual analysis, a concept is chosen for examination, and the analysis involves quantifying and tallying its presence (Palmquist et al., 2005). The focus here is on looking at the occurrence of selected terms within a text or texts, although the terms may be implicit as well as explicit (Palmquist et al., 2005).

Conceptual analysis begins with identifying research questions and choosing a sample or samples (Palmquist et al., 2005). Once chosen, the text must be coded into manageable content categories. The process of coding is basically one of selective reduction. By reducing the text to categories consisting of a word, set of words or phrases, the researcher can focus on, and code for, specific words or patterns that are indicative of the research question. This analysis is focused on the initial set of terms

"Software Quality Assurance", "Automated Testing Software", "Software Testing strategies" and "Web Based Applications" and identification of the context surrounding the discussion of each term.

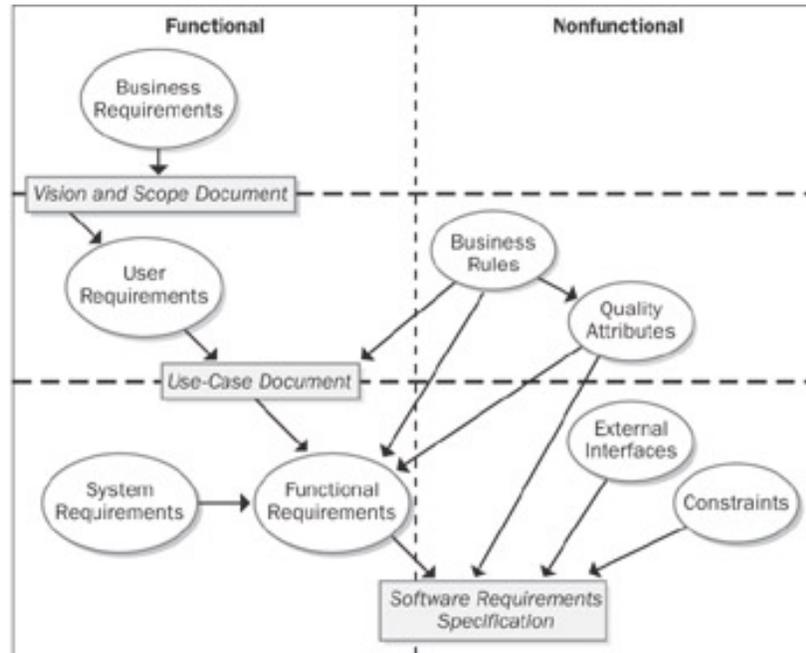
Table 5: Benefits Of Using Software Automated Tools is designed to categorize the results of the conceptual analysis, using the initial set of terms listed, i.e., "Software Quality Assurance", "Automated Testing Software", "Software Testing Strategies" and "Web Based Applications" in a way that reveals the context of the discussions in relation to potential benefits reported, as a result of using automated testing software. The hope of the researcher is that by producing a list of common benefits that the industry has found as a result of using automated testing software, readers of the study will more clearly see ways that software testing strategies and the use of automated testing software can increase software quality assurance in web-based applications.

This study is designed for business analysts (Cram, C & Heldman, W, 2004) who deal with Internet-based object oriented software applications (Lethbridge, T & Laganière, R, 2001). Other terms for business analyst are business systems analyst (Rittinghouse, 2004) and requirements analyst, (Wiegers, 2003). The analyst serves as the principal conduit through which requirements flow between the customer community and the software development team, as shown in Figure 1 (Wiegers, 2003).

Figure 1: The Role of the Analyst

Different books refer to different roles of analysts in the process life cycle, using terms such as "Requirements Analyst" (Wiegiers, 2001), "Test Analyst" (Watkins, 2001), and "QA Analyst" (Watkins, 2001). In companies such as UPS, the Business analyst can take on the role of all three positions.

The business analyst is responsible for gathering and writing requirements (a requirement is "anything that drives design choices" (Wiegiers, 2003)). The software development process starts by defining the requirements (Goldsmith, 2004). Software requirements include three distinct levels—business requirements, user requirements, and functional requirements (Wiegiers, 2003). Figure 2 from Wiegiers demonstrates the relationship between the three types of requirements (Wiegiers, 2003).

Figure 2: Three Types of Requirements

Once the requirements are gathered, the analyst will create business rules (Wiegiers, 2003). A business rule, "describes the operations and constraints that apply to an organization in achieving its goals. They are often collected as part of the Requirement gathering process to be used in Use cases". (http://en.wikipedia.org/wiki/Business_rules)

The results of the conceptual analysis are presented in the form of a table (see Appendix B) that presents the context of the discussions in relation to the initial set of terms coded, i.e., "Software Quality Assurance", "Automated Testing Software", "Software Testing Strategies" and "Web Based Applications". Then further analysis is conducted with focus on extracting potential benefits of using automated testing software to support software quality assurance in web-based applications. Benefits are framed into the table (see Table 5: Benefits Of Using Software Automated Tools) as the final outcome.

Significance of Study

According to Schwartz (2003), the increasing complexity of software development demands that developers use automated software testing tools. Today, software organizations invest more time and resources in analyzing and testing software as a unit rather than as independent entities (Li & Wu, 2004).

Software engineers have observed that writing testing code is as expensive and time consuming as developing the product itself (Li & Wu, 2004). To ensure software quality, organizations encourage software developers and testers to achieve objectives such as these:

- Locating the source of defects faster and more precisely
- Detecting bugs earlier in the software development life cycle
- Removing more defects before the product is released
- Improved testing tools can reduce the cost of software development and increase the quality of software (Li & Wu, 2004).

Automation testing software is software that tests application performance and functionality, (Hayes, Whiting, 2002) and aids developers in debugging and releasing applications faster (Pepe, 2000, p. 57). Without testing, bugs and defects will hinder performance and usability of the application (Leon, 2003). Automated testing software can enhance the return on investment of applications by performing GUI (Graphic User Interface) testing, test creation, test scripting, load testing and a host of other functions. (Unknown author, Client Server Computing, Jan 1997) Companies

are increasingly turning to automated testing solutions to gain confidence in their IT systems (Shea, 2000, Schwartz, 2003).

According to Shea (2000, p. 97), as businesses move to integrated web enabled applications, testing becomes crucial. Companies intent on transacting revenue generating business or offering enhanced customer services online are increasingly turning to automated testing solutions to gain confidence in their IT systems (Shea, 2000). Applications performance and transaction precision equate to the efficient business services (Shea, 2000). This leads to improved customer satisfaction (Shea, 2000).

Many IT managers have been challenged to prove the business value of implementing an automated testing solution (Shea, 2000). Automated tools are costly, as are the qualified staff and hardware resources necessary to operate and maintain them (Shea, 2000). Shea (2000, pg 136), states that what tends to most easily convince businesses of the need to establish formal automated test processes is a bad experience with an application providing poor service, or one that fails altogether. Software failures result in a substantial economic loss to the United States each year (Li & Wu, 2004). Approximately half of the losses occur within the software manufacturing industry (Li & Wu, 2004).

Limitations

The literature selected for this study covers the rapid changes in Internet technology and usage that shape today's website design practices. This time frame (1998 - 2005) excludes older research and practices that over time have been replaced by more current design practices.

Material was selected from academic, professional, and association literature and websites. Academic material provided practical and theoretical context for the study. Because academic material incorporates use of a rigorous research design, this material may have validity and credibility that may not be found in other sources.

Research focused on several areas related to "Software Quality Assurance", "Automated Testing Software", "Software Testing strategies" and "Web Based Applications". Databases searched for this study included: scholar.google.com, Google, and EBSCO Host Research Database. A leading website related to Quality Assurance, QAForums.com, was searched for this study. This site contains over 50 forums that cover almost every area in software testing and quality assurance. Forums exist in regard to software test tools available like WinRunner, QuickTest Pro and LoadRunner by Mercury Interactive, Rational Robot and SilkTest by Segue Software.

Books24x7, a SkillSoft Company, is a leading provider of web-based digital technical and business reference content, including thousands of digitized "best-in-class" reference books, research reports, documentation and articles. Books24x7's hosted web-based platform enables users to search, browse, read and collaborate with vast professional "Referenceware"

libraries assembled through its relationships with the world's top IT and business publishers. This includes, in the technical publishing arena, imprints like Microsoft Press, Osborne/McGraw Hill, Artech House and many others. In the business field, prominent publishing partners include AMACOM, ASTD Press, Berrett-Koehler, Harvard Business School Publishing, John Wiley, MIT Press, Oxford University Press and others. Prominent customers that have adopted Books24x7 solutions include: Bank of America, Lockheed-Martin and UPS. Privacy issues, although an integral part of any website and related marketing activity, are not specifically included.

In his book *Effective Software Testing*, Dustin (2003) provides a clarification of what types of testing tools are available for use throughout the various software development life cycle phases. For the purpose of this research, "automated testing software" refers specifically to the following type of testing tool:

GUI-Testing Tools (Capture/Playback) – Automates GUI tests by recording interactions with online systems, so they may be replayed automatically (Dustin, 2003, pg 162)

According to the Merriam Webster online dictionary, "benefit" is defined as "to be useful or profitable to" (<http://www.m-w.com/cgi-bin/dictionary>). For the purpose of this research, "benefits" refer specifically to a positive value that is derived from utilizing the automated testing software. Benefits do not refer to "payment or service provided for under an annuity, pension plan, or insurance policy" (<http://www.m-w.com/cgi-bin/dictionary>) According to QA

Labs, there are five key factors of quality in reference to Web Applications (QA Labs, 2000).

- Quality
- Reliability
- Recoverability
- Security
- Usability
- Performance

The presentation of "benefits" in the final table is based on these factors.

Definitions

Term	Definition
Application	A program used by an end-user, as opposed to one simply used by other programs (http://www.site.uottawa.ca:4321/oose/application.html)
Applications software	(also called <i>end-user programs</i>) includes database programs, word processors, and spreadsheets. Figuratively speaking, applications software sits on top of systems software because it is unable to run without the operating system and system utilities. (http://www.webopedia.com/TERM/a/application.html)
Automated testing	that which is performed, to a greater or lesser extent, by a computer. (http://en.wikipedia.org/wiki/Automated_testing)
Automated testing tool	A device that provides a mechanical or mental advantage in automated testing. (http://en.wikipedia.org/wiki/Automated_testing) (http://en.wikipedia.org/wiki/Tool)
Benefit	To be useful or profitable to. In this study, the term refers to a positive value that is derived from utilizing the automated testing software. (http://www.m-w.com/cgi-bin/dictionary) According to QA Labs, there are five key factors of quality in reference to Web Applications (QA Labs, 2000). <ul style="list-style-type: none"> • Quality • Reliability • Recoverability • Security • Usability • Performance The presentation of “benefits” in the final table is based on these factors.
Bug	is an error, flaw, mistake, failure, or fault in a computer program that prevents it from working correctly or produces an incorrect result. (http://en.wikipedia.org/wiki/Software_bug)
Business rule	describes the operations and constraints that apply to an organization in achieving its goals. They are often collected as part of the Requirement gathering process to be used in Use cases. (http://en.wikipedia.org/wiki/Business_rules)
Business analyst	A person who is responsible for identifying the business needs of their clients and stakeholders to help determine solutions to business problems. (http://en.wikipedia.org/wiki/Business_analyst)

Defect	A flaw in any aspect of the system including the requirements, the design or the code, that contributes, or may potentially contribute, to the occurrence of one or more failures. (http://www.site.uottawa.ca:4321/oose/defect.html)
GUI	An acronym for Graphical User Interface. (http://isp.webopedia.com/TERM/G/GUI.html)
Graphical User Interface	A program interface that takes advantage of the computer's graphics capabilities to make the program easier to use. Well-designed graphical user interfaces can free the user from learning complex command languages. On the other hand, many users find that they work more effectively with a command-driven interface, especially if they already know the command language. http://isp.webopedia.com/TERM/G/Graphical_User_Interface_GUI.html
Load Testing	The practice of modeling the expected usage of a software program by simulating multiple users accessing the program's services. Concurrently. (http://en.wikipedia.org/wiki/Load_testing)
Object Oriented	A popular buzzword that can mean different things depending on how it is being used. <i>Object-oriented programming (OOP)</i> refers to a special type of programming that combines data structures with functions to create re-usable objects (see under <i>object-oriented programming</i>). Object-oriented graphics is the same as <i>vector graphics</i> . Otherwise, the term <i>object-oriented</i> is generally used to describe a system that deals primarily with different types of objects, and where the actions you can take depend on what type of object you are manipulating. For example, an object-oriented draw program might enable you to draw many types of objects, such as circles, rectangles, triangles, etc. Applying the same action to each of these objects, however, would produce different results. If the action is <i>Make 3D</i> , for instance, the result would be a sphere, box, and pyramid, respectively. (http://www.webopedia.com/TERM/o/object_oriented.html)
Quality Assurance	The process of ensuring that the quality of a product or process is sufficient to meet the needs of the stakeholders (http://www.site.uottawa.ca:4321/oose/qualityassurance.html)
Requirement	A description of <i>what</i> a system should do. Systems may have from dozens to hundreds of requirements. (http://en.wikipedia.org/wiki/requirement) The <i>IEEE Standard Glossary of Software Engineering</i>

	<p><i>Terminology</i> (1990) defines a <i>requirement</i> as</p> <ol style="list-style-type: none"> 1. A condition or capability needed by a user to solve a problem or achieve an objective. 2. A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document. 3. A documented representation of a condition or capability as in 1 or 2. <p>(Wiegiers, K. (2003) <u>Software Requirements, Second Edition</u>, Microsoft Press)</p>
Script	<p>A computer program that automates the sort of task that a user might otherwise do interactively at the keyboard (http://en.wikipedia.org/wiki/Script_%28computer_programming%29)</p>
Software	<p>Computer instructions or data. Anything that can be stored electronically is software. The storage devices and display devices are hardware. Software is often divided into two categories: Systems software: Includes the operating system and all the utilities that enable the computer to function; and Applications software: Includes programs that do real work for users. For example, word processors, spreadsheets, and database management systems fall under the category of applications software. (http://www.webopedia.com/TERM/s/software.html)</p>
Software Development life cycle	<p>A framework for describing the phases involved in developing information systems. The cycle consists of the following phases:</p> <p>Analyze Design Implement Test Deliver Support</p> <p>Some popular models of a SDLC include the waterfall model, the spiral model, and the incremental build model. (http://en.wikipedia.org/wiki/Software_development_life_cycle)</p>
Software Testing	<p>A process used to identify the correctness, completeness and quality of developed computer software. Actually, testing can never establish the correctness of computer software, as this can only be done by formal verification (and only when there is no mistake in the formal verification process). It can only find defects, not prove that there are none. (http://en.wikipedia.org/wiki/Software_testing)</p>

Software Testing Tool	A device that provides a mechanical or mental advantage in software testing. (http://en.wikipedia.org/wiki/Software_testing) (http://en.wikipedia.org/wiki/Tool)
Strategy	A long term plan of action designed to achieve a particular goal. (http://en.wikipedia.org/wiki/Strategy)
Test Automation	The use of software to control the execution of tests, the comparison of actual outcomes to predicted outcomes, the setting up of test preconditions, and other test control and test reporting functions. (http://en.wikipedia.org/wiki/Test_automation)
Testing	An approach to verification that involves systematically executing software to detect defects (Fact Guru Object Oriented Software Engineering http://www.site.uottawa.ca:4321/oose/testing.html)
Test case	A set of conditions or variables under which a tester will determine if a requirement is partially or fully satisfied. It may take many test cases to determine that a requirement is fully satisfied. In order to fully test that all the requirements of an application are met, there must be at least one test case for each requirement unless a requirement has sub requirements. In that situation, each sub requirement must have at least one test case. (http://en.wikipedia.org/wiki/Test_case)
Time to market	The length of time it takes to get a product from idea to marketplace. (http://www.investorwords.com/4986/time_to_market.html)
Tool	A device that provides a mechanical or mental advantage in accomplishing a task. (http://en.wikipedia.org/wiki/Tool)
Use Case	A technique for capturing the potential requirements of a new system or software change. Each use case provides one or more <i>scenarios</i> that convey how the system should interact with the end user or another system to achieve a specific business goal. (http://en.wikipedia.org/wiki/Use_case)
Web Application	An application delivered to users from a web server over a network such as the World Wide Web or an intranet. (http://en.wikipedia.org/wiki/Web_application) This type of application has two very important characteristics: <ol style="list-style-type: none"> 1. A Web application is designed <i>from the start</i> to run in a Web-based environment. This means that the <i>hypermedia aspect</i> in terms of hypertext and multimedia in combination with

traditional application logic must be taken into account throughout the application lifecycle, which makes it different with respect to a conventional application.

2. A Web application is an application, not just a set of Web pages. In particular, this implies that it enforces the notion of a *session*, which differentiates it from the ordinary request-response Web paradigm. In this context, even a Web service that dynamically generates pages may not be considered a Web application. Think for example of a timetable service that, given desired departure and destination times and places, returns a set of pages containing the available trains and connections. In this case, there is no need for the service to maintain the notion of session, which means that, in our view, this is *not* a Web application, but just a Web-based service.

Feichtner, C., Finkelstein, A., Kappel, G., Retschitzegger, W., Savigni, A., Schwinger, W. (2003, Jan). Ubiquitous Web Application Development -A Framework for Understanding International Journal of Web Engineering and Technology, January 2003.
<http://www.schwinger.at/LIB/2001/ASE2001/ASE2001.pdf>

In this study, The term "Web Based Applications" refers to Internet-based object oriented software applications (Lethbridge, T & Lavalieres, R, 2001). The purpose of using the specific term is to narrow the focus of the paper to deal specifically with applications that are not mainframe based, but are used in a web-based environment (Schwinger et al., 2003). These applications have a Graphical User Interface (GUI), and are delivered to users from a web server over a network such as the World Wide Web or an intranet ([http://en.wikipedia.org/wiki/Web application](http://en.wikipedia.org/wiki/Web_application)).

Problem Area

There is a growing concern for software quality (Coffee, 2003). Presentation topics at the 2003 International Conference on Quality Software included "test case generations from software specifications", "checklists and models for estimating software costs", and "the challenges of defining and testing software quality in e-commerce environments" (Coffee, 2003). Gartner analyst Theresa Lanowitz says about testing and software, "Quality is more important with Web services than other types of applications, because you have all this loosely coupled software that comes together to form a system and work together in a perfect state. Testing will become a very tactical part of the software development cycle" (Hayes & Whiting, 2002)

Software testing is an integral, costly and time-consuming activity in the software development life cycle (Rankin, 2002). Testing and software quality assurance play critical roles in the successful implementation of any application (Mayer, J, 1998). Increased time-to-market pressures in recent years have resulted in a deterioration of the quality of software entering a system test phase (Asada & Yan, 1998). With deadlines tightening and workloads increasing, software engineers are constantly looking for ways to improve the process of software development (Van Tongeren, 2000). One way to maximize their efficiency is with automatic test case generation (Van Tongeren, 2000).

Automated testing is testing that is performed, to a greater or lesser extent, by a computer (<http://en.wikipedia.org>). Software is available on the market that can control the execution of tests, the comparison of actual

outcomes to predicted outcomes, the setting up of test preconditions, and other test control and test reporting functions

(http://en.wikipedia.org/wiki/Test_automation). There are many types of testing tools available for use throughout the various software development life cycle phases (see Table 1: Testing Tools) (Dustin, 2003, p. 159).

Table 1: Testing Tools (from Dustin, 2003, p. 162)

Type of Testing Tool	Description
Test Procedure Generators	Generate test procedures from requirements/design/object models
Code (Test) Coverage Analyzers and Code Instrumentors	Identify untested code and support dynamic testing
Memory-Leak Detection	Verify that an application is properly managing its memory resources
Metrics-Reporting Tools	Read source code and display metrics information, such as complexity of data flow, data structure, and control flow. Can provide metrics about code size in terms of number of modules, operands, operators, and lines of code.
Usability-Measurement Tools	User profiling, task analysis, prototyping, and user walk-throughs
Test-Data Generators	Generate test data
Test-Management Tools	Provide such test-management functions as test-procedure documentation and storage and traceability
Network-Testing Tools	Monitoring, measuring, testing, and diagnosing performance across entire network
GUI-Testing Tools (Capture/Playback)	Automate GUI tests by recording user interactions with online system, so they may be replayed automatically
Load, Performance, and Stress Testing Tools	Load/performance and stress testing
Specialized Tools	Architecture-specific tools that provide specialized testing of specific architectures or technologies, such as embedded systems

Companies are increasingly turning to automated testing solutions to gain confidence in their IT systems (Shea, 2000, p. 97). Automated testing

tools can enhance the testing effort (Dustin, 2003, p. 159). Automation can reduce the role of manual testing, and shorten the time to market for new applications (Hildreth, 2004). According to Shea (2000), utilizing an automated testing strategy can help deliver the products to market sooner with more accuracy and less user-found errors (p. 136).

Schwartz (2003) believes that the increasing complexity of software development demands that developers use automated software testing tools. Most importantly, as stated by Hildreth (2004), the desire for repeatability and accuracy is a reason organizations are moving to automate testing, both to uncover bugs and to ensure performance standards.

CHAPTER II – REVIEW OF REFERENCES

This section provides a review of the key references used to define the framing, purpose, problem area and research method of the study. The sources are categorized based upon their pertinence to the central topics of the study, which are:

- "Software Testing Strategies"
- "Software quality assurance"
- "Automated testing software"
- "Web Based Applications".

Within these topics, each reference is annotated according to three points:

1. The specific content used in this study;
2. How this content is used as support within the following parts of the study:
 - The purpose of the study including the studies significance, scope and limitations
 - The problem area of the study
 - The method of the study
3. The criteria used for selection, including credibility and pertinence.

References Related to Software Testing Strategies

Copeland, L., (2004). A Practitioner's Guide to Software Test Design. Artech House

This text focuses only on software test design, and the more practical aspects of testing, specifically test case design. It describes a set of key test design strategies that improve both the efficiency and effectiveness of software testers. The audience for this book includes quality assurance and process improvement engineers who are charged with defining and improving their software testing process. This source is used in the Full Purpose section to provide a definition of what "testing" is.

The text's author is an internationally known consultant in software testing, with over 30 years of experience as an information system professional. He has held a number of technical and managerial positions with commercial and nonprofit organizations in the areas of software development, testing, and process improvement. He has taught seminars and consulted extensively throughout the United States and internationally.

Mayer, J. (1998, May). The right way to view testing. Software Magazine, Supplement, Vol 18, Issue 7, p. 3 - 7

In this article Mayer argues that software test managers have to sell their function as an integral part of the software development process in order for them to earn the respect and funding they deserve. He identifies a relationship between testing and software quality assurance, and highlights the use of automated testing tools to improve the quality of the software being tested.

This source is used in the Brief Purpose to establish a relationship between testing and software quality assurance and stress the importance of testing in the software development life cycle. The source is also used in the Full Purpose to elaborate on the relationship between testing and software quality assurance.

References Related to Software Quality Assurance

Coffee, P. (2003, Oct). Assuring software quality.
eWeek, Vol. 20 Issue 43, p41, 2p

This paper is a result of findings that were presented at the International Conference on Quality Software, in Dallas, Texas on November 6 and 7, 2003, and deals specifically with the apparent IT community concerns in relation to Software Quality. This source is used in the Problem section to establish the importance of Software quality.

The author of this article is the Technology Editor for eWeek magazine. He served as Technology Editor at PC Week, the national newspaper of enterprise computing. He serves as an internal consultant to the editorial staff and as the technical liaison between PC Week and its advisory panel of corporate information technology architects. He is the author of "How To Program Java" (ZD Press) and "Peter Coffee Teaches PCs" (Que) and has received many professional awards for his contributions to PC Week's editorials, print and on-line columns, and product reviews. He is a frequent speaker at industry conventions and conferences, and often appears on ZDTV programs including Silicon Spin and Big Thinkers. Before becoming a full-time writer in 1989, Peter held project management and technical positions with Exxon and The Aerospace Corporation.

References Related to Automated Testing Software

Gao, J.Z., Tsao, H.S., Wu, Y. (2003). Testing and Quality Assurance for Component-Based Software, Artech House

This book covers concepts and fundamentals for component-based software testing, validation methods for software components and component-based programs, strategies for integration, regression, and performance testing of component-based software.

This source is used in the Brief Purpose to define Software test automation. The source is also used in the Full Purpose to define software quality assurance. As described below, each of the text three authors has a background in software development and testing.

Jerry Zeyu Gao is an associate professor in the Department of Computer Engineering at San José State University. He received an M.S. and a Ph.D. from the University of Texas at Arlington. Dr. Gao has more than 10 years of industry development experience and management experience with software development, testing, and Internet applications. Dr. Gao joined the Department of Computer Engineering at San José State University in 1998. His research areas include software engineering, component-based software engineering, and Internet computing, e-commerce, and wireless Web applications. He has published widely in IEEE/ACM journals, magazines, and international conferences. He has published more than 40 technical papers and has written a book, *Object-Oriented Software Testing* (IEEE Computer Society Press, 1998).

Gao teaches programming in C/C++/Java, software engineering I and II, data structure and algorithms, object-oriented technology, and software

testing and quality assurance. In addition, he has developed and taught a number of new courses on emerging technology, such as Design of Web-Based Application Systems, Design of E-Commerce Systems, and Design of Mobile-Based Software Systems.

H.-S. Jacob Tsao has been an associate professor of industrial and systems engineering at San José State University since 1999 and has taught courses in quality assurance, reliability engineering, information engineering, operations research, and statistics. He has numerous technical publications, including more than 30 refereed journal papers and a book, *Entropy Optimization and Mathematical Programming* (Kluwer, 1997). He received a B.S. in applied mathematics from the National Chiao-Tung University in Taiwan in 1976, an M.S. in mathematical statistics from the University of Texas at Dallas in 1980, and a Ph.D. in operations research from the University of California at Berkeley in 1984. Dr. Tsao has worked for Consilium Inc. (currently an Applied Materials company) as a software development engineer on computer-aided manufacturing, and for AT&T Bell Laboratories and Bell Communication Research as a systems engineer on large-scale software systems designed to automate circuit provisioning and capacity expansion planning. He joined the Institute of Transportation Studies of University of California at Berkeley in 1992 and researched concept development and evaluation for intelligent transportation systems.

Ye Wu received a B.S. and an M.S. in computer science from Shandong University, China, in 1993 and 1996, respectively. He received a Ph.D. in computer science from the State University of New York at Albany in

2000. Dr. Wu joined the faculty of George Mason University in 2000 as an assistant professor in the Department of Information and Software Engineering. His research interests include software testing, software maintenance, and software architecture analysis. Currently, his research focuses on testing and maintaining component-based and Web-based software.

Dustin, E., (2003). Effective Software Testing: 50 Specific Ways to Improve your testing. Addison-Wesley

This book focuses on the key aspects of software testing by exploring fifty critically important best practices, pitfalls, and solutions. It emphasizes a need to integrate testing into all phases of the software development lifecycle. Automated testing is one of the topics covered.

This source is used in the limitation section to define the type of automated testing software that is being referred to in this research. The source is also used in the Problem Area section to display the various types of automated testing software that is available.

The author is an acknowledged expert in software engineering and testing practices. She is the Testing Manager at the Bureau of National Affairs and is coauthor of *Automated Software Testing* (Addison-Wesley, 1999), the first book to address this critical topic. An acknowledged expert and Software Quality Assurance certified test engineer, she has worked on automated test tool rollouts for numerous companies and projects. She is also president of the Washington, D.C. area Rational Testing User Group

Hayes, M., & and Whiting, R (2002, Sept). Testing Tools Are Key to Web Services' Success. InformationWeek, no.904, 2 Sept. 2002. p. 56-60. Journal Paper

This article focuses on providing tips for the proper operation and performance of web services. The resource is used in the Brief Purpose to further define automated testing tool, and is later used in the Significance to define automated testing tool.

Rick Whiting is an editor for Information Week magazine. An article written by him appears weekly in the magazine. Mary Hayes is a writer for Information Week magazine.

Hildreth, S. (2004, Oct). Automating the 40 Monkeys. Computerworld; Vol. 38 Issue 40, p24, 2p.

The article focuses on the use of software tools for pre-deployment application testing by the business enterprises. Vendors such as Mercury, Segue Software, Compuware Corp., IBM's Rational unit, RadView Software Ltd. and Empirix Inc. offer products for testing client's server and Web-based applications, as well as mainframe and other legacy systems. Most of these products provide an interface for programming test scripts, as well as a capture-replay tool or visual scripting option for non-technical users.

The source is used in the Problem Area section to illustrate the use of automated testing software and the reduction of manual testing time and reduction of time to market.

The author of this article is a freelance writer specializing in enterprise software technology trends. She has followed the computer industry since 1986, when she began writing new product news for Cahners Business Information's Business Computer Systems magazine. More recently, she

served as the executive editor for ebizQ, an online publication covering e-business integration trends, and was an editor for Computer Publishing Group's two publications: WebServer Online Magazine and Server/Workstation Expert, a monthly magazine covering the Unix market.

Li, K., and Wu, M. (2004). Effective Software Test Automation: Developing an Automated Software Testing Tool, Sybex

This text illustrates how to build a fully automated testing tool for immediate use for software development projects, and provides expert guidance on deploying it. The source is used in the Significance of Study section to explain the importance of analyzing and testing software and what organizations do to ensure Software Quality and to emphasize the cost of writing code vs. developing the product. It is also used to list goals that organizations encourage software developers and testers to aim for to increase software quality assurance.

Kanglin Li has worked as a software design engineer for Agilent Technologies and Communications Data Services, and has served as Assistant Professor at North Carolina A&T University. He is interested in techniques for automating key software development tasks. Mengqi Wu is a system engineer at Lucent Technologies' Bell Labs. She holds degrees in computer science and law.

Pepe, M. (2000, Jul). Apps Testing Speeds Time To Market. Computer Reseller News, Issue 904, p57, 2p

This article focuses on the efforts of programmers to develop automated testing for Web application software, and lists challenges faced by developers in testing a product. This source is used in the Brief Purpose to

explain the value of using automated software testing tools to increase software quality assurance for Internet-based applications.

Michele Pepe edits section stories and writes for *CRM's* Special Reports. Since joining the staff in 1994, she has taken on a number of roles--from copy editor to Special Reports manager--and has reported and written on myriad topics, including custom systems, peripherals, POS, smart cards and storage.

Before joining CMP Media, Pepe worked for *PC Magazine*, where she was a copy editor and staff writer. Pepe graduated from Cornell University in 1991.

Rankin, C. (2002). The Software Testing Automation Framework. IBM Systems Journal, Vol. 41 Issue 1, p126, 14p

This paper explores the opportunities for reuse and automation in one test organization, describes the shortcomings of potential solutions that are available "off the shelf," and introduces a new solution for addressing the questions of reuse and automation, the Software Testing Automation Framework (STAF), a multiplatform, multi-language approach to reuse. It is based on the concept of reusable services that can be used to automate major activities in the testing process. A case study is discussed regarding how to automate a resource-intensive test suite used by an actual testing organization within IBM.

This source is used in the Brief Purpose to highlight the importance of software testing in the software development life cycle, and is used in the Problem section to illustrate the time saving ability of testing automation.

Charles Rankin is an advisory software engineer in the IBM Austin Development Laboratory. He graduated with a BS degree in electrical engineering from the University of Florida in 1993, then joined IBM in Austin. He has worked extensively with IBM's PC oriented operating systems and networking products. He was the system test lead for IBM's Directory and Security Server for OS/2 and IBM's OS/2 WARP Server for e-Business. He is currently the lead developer for STAF.

Schwartz, K. (2003, Sept). Tools Automated Software Testing InformationWeek, Issue 955, p55, 3p

This article provides a brief history of the software industry and the use of testing tools in the 90s, and focuses on the development of software testing tools by the business enterprises in the U.S. It provides an identification of three types of testing tools in the software development phase.

This source is used in the Brief Purpose to establish the research focus of the paper – “the increasing complexity of software development demands that developers use automated software testing tools”. It is used in the Full Purpose section to elaborate on the complexity of business applications and used in the Significance of Study section to reinforce the claim that companies are turning to Automated software tools as a method to testing more complex applications.

Karen D. Schwartz is a freelance writer specializing in business and technology, and is based in the Washington, DC area.

Shea, B. (2000, July). Software Testing Gets New Respect. InformationWeek, Issue 793, p97, 6p.

This article focuses on the importance of automated software testing and highlights the increasing application of automated software testing by business enterprises. It provides an estimated market for automated software quality tools, and establishes factors that influence the adoption of automated testing solutions.

This source is used in the Significance of Study section to establish the claim that companies are increasingly turning to automated testing solutions to gain confidence in their IT systems. Shea stresses the importance of testing as more businesses move to integrated web applications.

Billie Shea is director of research and reporting services with Newport Group, Inc., an IT trends research and reporting firm based in Barnstable, MA. She has been involved in the information technology industry for the past ten years. In addition to conducting extensive industry, market and technology analysis for Newport Group, Inc., Billie directs the development of white papers, research notes and custom research, with particular emphasis on quality assurance practices and tools, application development practices and tools and performance management. In addition to Newport Group published works, she has been a regular contributor to multiple industry publications such as Software Testing & Quality Engineering, Information Week, Application Development Trends, Software Magazine, and Enterprise Systems Journal among others; and has been quoted in numerous others. She contributed to a recent book on Web application testing titled "Making E-Business Work", and is currently in the process of contributing to the book's

second edition. With a background in computer science and Industrial/Organizational HR, she holds a Bachelors degree from the University of Connecticut.

Van Tongeren, T. (2000, Apr). A Tester's Tale.

Software Magazine: Vol. 20 Issue 2, p46, 3p.

This article outlines an experience in a rapid application development environment and the benefits that resulted by automating code generation and test case generation. It explains the benefits of automatic test case generation. This source is used in the Problem section to explain the need to improve the process of software development and how automated testing can assist in that.

Tim Van Tongeren is a software tester with articles published in the Journal of Software Testing Professionals and Software Testing and Quality Engineering magazine. He is currently serving on the Board of Directors for the Association for Software Testing.

Watkins, J., (2001). Testing IT: An Off-the-Shelf Software Testing Process. Cambridge University Press

This book provides guidance to workers in the field of software testing for researching or setting up a software testing process within organizations. This source is used in the Brief Purpose section to define the term "Automated Testing tool". It is used in the Full Purpose section to define the terms "Test Analyst" and "QA Analyst". This source is used in the Method section to define the term "Automated Test Software".

John Watkins holds Masters Degrees in both Computer Science and Object-Orientation, and has over 23 years experience in the field of software

development, with some 18 years in the field of software testing. He is a Fellow of the British Computer Society and a Chartered IT Professional. During his career as a testing professional, Watkins has been involved at all levels and phases of testing, and has provided high level testing consultancy, training and mentoring to numerous Blue Chip Companies. He currently works for the IBM Software Group in the UK.

Watkins is a regular presenter at international testing conferences and events, having recently addressed the ICS Test 2004 conference in Dusseldorf, as well providing the keynote address at the most recent Ohjelmistotestaus conference in Helsinki.

References Related to Web Based Applications

QA Labs, (2003). ["The Living Creature" – Testing Web Applications](http://www.qalabs.com/resources/articles/thelivingcreature.pdf)
<http://www.qalabs.com/resources/articles/thelivingcreature.pdf>

This white paper, obtained from <http://www.qalabs.com>, was presented to the Vancouver chapter of the IEEE/ASQ at the May 10, 2000 evening seminar, and at multiple seminars for IS professionals in 2000. This paper focuses on what a Web application are, differences between web applications and traditional business transaction applications, and what the challenges are in testing a web application. Factors that quantify quality assurance are explained in detail. Testing strategies towards web applications are also suggested and thoroughly explained.

This resource is used in the Full Purpose section to illustrate Web application architecture. It is used in the Problem section to further establish the issue of Quality in terms of Web applications, and establishes five key factors of quality in reference to Web Applications. These factors are:

quality, reliability, recoverability, security, usability and performance. The presentation of “benefits” in the final table is based on these factors established by QA Labs.

Founded in 1999, QA Labs has become the largest independent software testing and quality assurance provider in Canada.

Feichtner, C., Finkelstein, A., Kappel, G., Retschitzegger, W., Savigni, A., Schwinger, W. (2003, Jan). Ubiquitous Web Application Development - A Framework for Understanding International Journal of Web Engineering and Technology, January 2003.

This paper is a result of findings that were presented at various world symposiums in 2002, and deals specifically with what a “Web based application” is. This source is used in the Full Purpose and Definitions section to define the term “Web based application”, and narrows the scope of the paper to deal specifically with Web based applications.

This work was partially funded by UWA (Ubiquitous Web Applications), an EU funded Fifth Framework Program project (IST-2000-25131) that the authors are carrying on in co-operation with a number of academic and industrial partners from six European countries. The work described in this paper is intended as an initial contribution to the project.

Laganière, R. & Lethbridge, T. (2001). Object Oriented Software Engineering: Practical Software Development using UML and Java, McGraw Hill.

The book is designed for students specializing in software engineering. It should be useful, however, for anyone who has a background in programming and wants to learn practical software engineering techniques. A major feature of the book is fully worked examples, implemented in Java. This knowledge base provides access to the key facts found in the book.

This document is the source for the definition of “web based application” found in the Brief Purpose section as well as the Definitions section.

Dr. Timothy C Lethbridge is Associate Professor and **Dr. Robert Laganière** is Assistant Professor in the School of Information Technology and Engineering at the University of Ottawa, Ontario.

[This page intentionally left blank.]

CHAPTER III – METHOD

The primary research method for this study is a literature review (Leedy & Ormrod, 2001) of sources. Literature review enables the researcher to collect, evaluate, analyze, and organize literature in order to conduct a study (Leedy & Ormrod, 2001). Literature is collected from materials published between December 1997 and April 2005. Leedy and Ormond (2001) suggest methods in which to identify subproblems within the main problem.

In determining the criteria for keyword search, the following problems were identified and broken down into subproblems:

1. How does implementing a testing automation strategy improve quality assurance for web-based applications?
2. How does automating testing improve quality assurance?

From these questions, additional questions were identified:

1. What is *testing automation*?
2. What is *quality assurance*?
3. What is a *web application*?

From these questions, the keywords and phrases were identified. Definitions were collected developed relating to the following terms:

- "Testing automation" refers to software that controls the execution of tests, the comparison of actual outcomes to predicted outcomes, the setting up of test preconditions, and other test control and test reporting functions.

- "Automated testing" refers to testing that is performed, to a greater or lesser extent, by a computer.
- "Automated test software" refers to software tools that automate the testing process in order to reduce effort, timescales, and cost in the testing of software (Watkins, 2001).
- "Quality assurance" refers to the process of ensuring that the quality of a product or process is sufficient to meet the needs of the stakeholders
- "Web based applications" refers to an application delivered to users from a web server over a network such as the World Wide Web or an intranet.

Data Collection

Literature searches were performed using the EBSCO Host Research Database to collect the data. The following terms were used:

- "software quality assurance"
- "automated testing Software"
- "software testing strategies"
- "web based applications".

Additional searches were performed on the following terms:

- "software test automation" & "web applications" & "quality Assurance"
- "automation testing software" & "quality Assurance"
- "automated testing tool" & "web applications"

Additional searches were performed utilizing the website

<http://www.scholar.google.com>. The website Books24x7

(<http://www.books24x7.com>) is a web-based digital technical and business reference content site, containing thousands of digitized "best-in-class" reference books, research reports, documentation and articles. Using this site, specific books related to "automated testing", "testing automation" and "software quality assurance" were found. Using the following search criteria:

- "software quality assurance"
- "automated testing software"
- "software testing strategies"
- "web based applications".

The following six results were selected, that also matched the Limitations set for this study:

Wieggers, K. (2003) Software Requirements, Second Edition, Microsoft Press

Cram, C., Heldman, W. (2004), IT Project+ Study Guide, Second Edition, Sybex

Gao, J.Z., Tsao, H.S., Wu, Y. (2003) Testing and Quality Assurance for Component-Based Software, Artech House

Li, K., and Wu, M. (2004) Effective Software Test Automation: Developing an Automated Software Testing Tool, Sybex

Watkins, J., (2001) Testing IT: An Off-the-Shelf Software Testing Process. Cambridge University Press

Copeland, L., (2004). A Practitioner's Guide to Software Test Design. Artech House

A qualitative review (Creswell, 1994) is performed of the sources gathered in the data collection stage to group the material by each of the terms defined in the data collection stage. The goal of this stage is to organize the collected information in anticipation of the data analysis process

in a way that represents the focus of the study (Krippendorff, 2004, pg. 349-353; Leedy & Ormrod, 2001, pg. 156-157).

Data Analysis

Content analysis is “a research technique for making replicable and valid inferences from texts (or other meaningful matter) to the contexts of their use” (Krippendorff, 2004, pg. 18). A specific form of content analysis, known as conceptual analysis, is selected as the data analysis strategy for use in this study (Palmquist et al., 2005). In conceptual analysis, a concept is chosen for examination, and the analysis involves quantifying and tallying its presence (Palmquist et al., 2005). The focus is to look at the occurrence of selected terms within a text or texts (Palmquist et al., 2005).

By reducing the text to categories consisting of a word, set of words or phrases, the researcher can focus on, and code for, specific words or patterns that are indicative of the research question (Palmquist et al., 2005).

There are 8 steps in the Conceptual model (Palmquist et al., 2005).

Step 1: Decide the level of analysis.

The level of analysis must be decided upon (Palmquist et al., 2005).

Literature is coded by the following initial set of phrases:

- “software quality assurance”
- “automated testing software”
- “web Based Applications”.

Step 2: Decide how many concepts to code for.

The number of different concepts to code for must be determined (Palmquist et al., 2005). Flexibility is needed to code for phrases and terms

similar to “Software quality assurance”, “Automated testing software”, and “Web based applications”. The following additional terms are coded under the original phrases:

- “Software quality assurance”
 - “quality assurance”
 - “software quality”
- “Automated testing Software”
 - “automated testing tool”
 - “testing automation software”
 - “automated testing”
 - “testing automation”
 - “automation testing”
- “Web Based Applications”.
 - “web applications”
 - “java applications”
 - “object oriented applications”

Step 3: Decide whether to code for existence or frequency of a concept.

Will the number of concepts be coded for existence or frequency (Palmquist et al., 2005)? The concepts are coded by the occurrence of the specific terms along with the context of the discussion of each of these. Additional similar terms that emerge are noted, along with the relevant context.

Step 4: Decide on how you will distinguish among concepts.

Will the concepts be coded exactly as they appear, or will they be recorded as the same even when they appear in different forms (Palmquist et al., 2005)? The literature is coded by similar concepts, beginning with the most well known options described below:

- "Software quality assurance"
 - "quality assurance"
 - "software quality"
- "Automated testing Software"
 - "automated testing tool"
 - "testing automation software"
 - "automated testing"
 - "testing automation"
 - "automation testing"
- "Web Based Applications".
 - "web applications"
 - "java applications"
 - "object oriented applications"

Step 5: Develop rules for coding your texts.

Translation rules are created that allow (agrees with rules) streamlining and organizing of the coding process so that coding is for exactly what is needed (Palmquist et al., 2005). This is to insure consistent coding throughout the text (Palmquist et al., 2005). Coding begins with just the initial list of phrases, "software quality assurance", automated testing

software", and "web based applications". Slight variations are covered under the umbrella of the specific term. An example of this situation is "automated", which is interpreted to mean the same as "automation".

Step 6: Decide what to do with "irrelevant" information.

The researcher must decide whether irrelevant information should be ignored (as Weber, 1990, suggests), or used to reexamine and/or alter the coding scheme (Palmquist et al., 2005). Any literature that is not related to "Web based applications" is ignored. The term "Web Based Applications" refers to Internet-based object oriented software applications (Lethbridge, T & Lavalieres, R, 2001). The purpose of using the specific term is to narrow the focus of the paper to deal specifically with applications that are not mainframe based, but are used in a web-based environment (Schwinger et al., 2003). These applications have a Graphical User Interface (GUI), and are delivered to users from a web server over a network such as the World Wide Web or an intranet (http://en.wikipedia.org/wiki/Web_application).

Step 7: Code the texts.

Coding the text is done either by hand, i.e. reading through the text and manually writing down concept occurrences, or through the use of various computer programs (Palmquist et al., 2005). Literature is read and coded by hand and transferred to a spreadsheet.

Step 8: Analyze your results.

Once the contextual discussion related to the initial coding phrases has been identified, it is further analyzed in relation to interpreting the "benefits" of using an automated testing strategy. For the purpose of this research,

“benefits” refer specifically to a value that is derived from utilizing the automated testing software. Benefits are identified in terms of five key factors of quality in reference to Web Applications, as described by QA Labs and presented in the Conclusion of this study.

Data Presentation

The data analysis results drawn from the conceptual analysis of the selected sources are framed from the standpoint of benefits to business analysts and presented in a form of a table that displays the benefits of utilizing an automated testing tool for web-based applications (see Table 5: Benefits Of Using Software Automated Tools). A template for this table is presented below (see Table 2):

Table 2: Template: Benefits Of Utilizing An Automated Testing Tool

Quality Factor	Automated Testing
Reliability	
Recoverability	
Security	
Usability	
Performance	

The goal of Table 5: Benefits Of Using Software Automated Tools is to list the advantages of using an automated testing tool based on the five quality factors established by QA Labs versus performing manual testing. These factors are: quality, reliability, recoverability, security, usability and performance. Benefits of using automated software-testing tools and the impact on software quality assurance are presented and described for

business analysts in the Conclusion chapter of this study. Table 5 is intended to demonstrate that a relationship exists between software testing strategies that utilize automated software testing tools and increased software quality assurance for Internet-based applications (Mayer, 1998; Pepe, 2000).

[This page intentionally left blank.]

CHAPTER IV – ANALYSIS OF DATA

This chapter is a report of the results of the content analysis conducted to define the benefits of utilizing an automated testing tool for web-based applications. Twenty-Five sources are analyzed, listed in Appendix A – Sources Used In Content Analysis. What follows is a summary of the coding process as applied, based on the eight-step description of conceptual analysis from Palmquist, et.al. (2005), and a reporting of the results. The primary reporting of the results of content analysis is presented in Appendix B – Contextual Discussions Related to Software Quality Assurance, Automated Testing Software and Web Based Applications. A summary discussion of the content in Appendix B can be found in this chapter.

Step 1: Decide the level of analysis.

Literature is coded by the following initial set of phrases:

- “Software quality assurance”
- “Automated testing software”

Step 2: Decide how many concepts to code for.

Coding incorporates ten concepts (see Table 3: List of Coding Phrases and Associated Terms). The purpose of Table 3 is to show the associated terms that are equated with the phrases “Software quality assurance”, “Automated testing software” and “Web based applications”.

Table 3: List of Coding Phrases and Associated Terms

Coding Phrases	Associated Terms
Software quality assurance	"quality assurance" "software quality"
Automated testing Software	"automated testing tool" "testing automation software" "automated testing" "testing automation" "automation testing"
Web Based Applications	"web applications" "java applications" "object oriented applications"

Step 3: Decide whether to code for existence or frequency of a concept.

The concepts are coded by the occurrence of the specific terms along with the context of the discussion of each of these.

Step 4: Decide on how you will distinguish among concepts.

The literature is coded by similar concepts (see Table 3: List of Coding Phrases and Associated Terms).

Step 5: Develop rules for coding your texts.

Table 4: Full List of Coding Terms presents the full list of associated terms covered under each of the key coding phrases.

Table 4: Full List of Coding Terms

Coding Phrases	Full List of Associated Terms
Software quality assurance	"quality assurance" "software quality" "QA" "Quality of software development"
Automated testing Software	"automated testing tool" "software testing tool" "testing automation software" "automated testing" "testing automation" "automation testing" "automation"
Web Based Applications	"web applications" "web apps" "java applications" "object oriented applications"

Coding begins with just the initial list of phrases, “software quality assurance”, “automated testing software”, and “web based applications”. Slight variations are covered under the umbrella of the specific term. An example of this situation is “automated” which is interpreted to mean the same as “automation”.

Step 6: Decide what to do with "irrelevant" information.

Any literature that is not related to “Web based applications” is ignored. The term “Web Based Applications” refers to Internet-based object oriented software applications (Lethbridge, T & Lavalieres, R, 2001).

Step 7: Code the texts.

Literature is read and coded by hand and transferred to a spreadsheet. A list of the 25 sources that were included in the coding is presented in Appendix A: Sources Used in Conceptual Analysis.

The overall goal of the conceptual analysis is to reveal contextual discussion in the literature, related to the initial three coding phrases (i.e. software quality assurance, automated testing software, and Web based applications), in relation to quality assurance. The resulting data set (which is twenty-seven pages in length) is presented in Appendix B – Contextual Discussions Related to Software Quality Assurance, Automated Testing Software and Web Based Applications. Appendix B consists of a report listing all instances of the keywords that occur in the selected references and the context in which they are being used.

Step 8: Analyze your results.

Additional review of Appendix B is focused on identification of issues concerning the benefits of using automated testing. For the purpose of this research, “benefits” refer specifically to a value that is derived from utilizing the automated testing software. In particular, benefits are identified in relation to five key factors of quality in reference to Web Applications, as described by QA Labs. These factors are: quality, reliability, recoverability, security, usability and performance. Benefits of using automated software-testing tools and the impact on software quality assurance are presented and described for business analysts in the Conclusion chapter of this study.

Once noted, benefits are recognized and reframed and presented in a form of a table (see Table 5: Benefits Of Utilizing An Automated Testing Tool) that displays the benefits of utilizing an automated testing tool in terms of the impact on software quality assurance. Benefits are described for business analysts in the Conclusion chapter of this study.

CHAPTER V – CONCLUSION

The data analysis results drawn from the conceptual analysis of the selected sources are presented in the twenty-seven page Appendix B. The results of the coding are re-framed from the standpoint of benefits to business analysts and presented in a final table (see Table 5: Benefits Of Utilizing An Automated Testing Tool). Benefits refer to ways of utilizing an automated testing tool for web-based applications in order to gain improved quality assurance. The goal of Table 5: Benefits Of Using Software Automated Tools is to list the advantages of using an automated testing tool based on the five quality factors established by QA Labs versus performing manual testing. Benefits are classified in terms of five key factors of quality in reference to Web Applications, as described by QA Labs. These factors are: quality, reliability, recoverability, security, usability and performance. The hope of the researcher is that this process will more clearly demonstrate ways in which software testing strategies and the use of automated testing software can increase software quality assurance in web-based applications. Doing this is intended to demonstrate that a relationship exists between software testing strategies that utilize automated software testing tools and increased software quality assurance for Internet-based applications (Mayer, 1998; Pepe, 2000).

Table 5: Benefits Of Using Software Automated Tools

Quality Factors	Automated Testing Benefits	Source
Reliability	Testing can be done faster than manual testing <ul style="list-style-type: none"> • Repeatability to identify regressions immediately • Insulation from human factors, such as boredom or carelessness • Improved audit trail to identify bugs • Shorter residence time of bugs in the code, reducing the manual effort of development and test engineers • Ability to test code in live development 	Pottage, D. (2001, Nov)
Reliability	<ul style="list-style-type: none"> • Removes much of the complexity of manual testing by emulating large user populations • Coordinates the interactions of different applications, databases, browsers, clients, servers and networks; • Builds test suites that can be adapted incrementally as systems evolve. 	Langley, N. (2003, Sept)
Reliability	<ul style="list-style-type: none"> • Developer-oriented tools are used to test specific pieces of the developer's application code independent of other units of code within the same app. They're especially useful in testing large applications where different developers are working on different parts of the app. 	Schwartz, K. (2003, Sept)
Reliability	<ul style="list-style-type: none"> • Automation of execution related activities offer another potential source of savings in the testing process. 	Rankin, C. (2002)
Recoverability	<ul style="list-style-type: none"> • Enhanced fault tolerance and automatic error recoverability 	Li, K., and Wu, M. (2004)
Security	<ul style="list-style-type: none"> • Easy portability with regard to installation, uninstallation, adaptability, and security 	Li, K., and Wu, M. (2004)
Usability	<ul style="list-style-type: none"> • Accurate functionality, reliability, interoperability, and compliance • An interface that is user friendly and easy to learn and operate 	Li, K., and Wu, M. (2004)
Usability	<ul style="list-style-type: none"> • Functional testing tools, usually used by a company's quality assurance department, test that the code being developed is acting as expected. • All tools in this category have scripting capabilities, letting developers modify what they've captured to test additional items. • One capability of a functional testing 	Schwartz, K. (2003, Sept)

Quality Factors	Automated Testing Benefits	Source
	<p>tool is capture/replay, which runs sample tests against a working version of a program, capturing the activity it generates.</p> <ul style="list-style-type: none"> • During the playback phase, developers can see whether they're getting the expected results. • Another subset of functional testing tools is regression testing, which tests whether a specific function continues normally after other changes have been made to the code 	
Usability	<ul style="list-style-type: none"> • The Parasoft SOAPtest tool that ABN Amro is using measured three main areas of quality testing: Functionality (how well the program works) 	Hayes, M., & Whiting, R (2002, Sept)
Performance	<ul style="list-style-type: none"> • Developer-oriented tools address memory leaks and other early performance problems. • Load testing. Also called stress or performance testing, these tools test what happens to the code when multiple users access the application simultaneously. Load testing is more critical in a Web environment, because it can be difficult to predict the volatility of the load change. 	Schwartz, K. (2003, Sept)
Performance	<ul style="list-style-type: none"> • The Parasoft SOAPtest tool that ABN Amro is using measured three main areas of quality testing: Functionality (how well the program works), load (how it performs under strain), and regression (whether any code changes result in problems). 	Hayes, M., & Whiting, R (2002, Sept)
Performance	<ul style="list-style-type: none"> • The desire for repeatability and accuracy is one reason organizations are moving to automate testing, both to uncover bugs and ensure that they meet performance standards 	Hildreth, S. (2004, Oct)

After reviewing the data presented in Table 5, common themes are noted (see Table 6: List of common themes in the data analysis).

Table 6: List of common themes in the data analysis results

Theme
Importance of Software quality
Importance of Automated testing
Software Testing market
When should automated testing be considered
How should automated testing be utilized
Type of automated testing tools available
Benefits of using automated testing
Challenges of utilizing automated testing
Examples of companies utilizing an Automated testing strategy

From these common themes, a number of key ideas are presented for business analysts.

Although they are often mistakenly referred to as one and the same, software quality assurance and testing play critical roles in the successful implementation of virtually any application (Asada & Yan, 1998). To ensure software quality, organizations encourage software developers and testers to achieve objectives such as these (Li & Wu, 2004):

- locating the source of defects faster and more precisely;
- detecting bugs earlier in the software development life cycle; and
- removing more defects before the product is released.

Quality assurance engineers are being asked to take on more complex projects than ever before (McNaughton, 2005). Manual testing is becoming a less viable alternative, and integration with the overall design processes and tools will prove necessary to keep pace in testing these complex current and future systems (Mats, 2002). More developers are using automated software

testing tools than ever before, and for good reason -- the increasing complexity of software development demands it. It pays to understand the capabilities available - now and in the future - in this critical area. (Schwartz, 2003) For mission-critical applications, automated testing is usually well worth the effort, given the consequences of failure (Hildreth, 2004).

Software test automation is one important and effective means to reduce manual testing effort and cost and to speed up and improve the quality of a software test process (Gao & Tsao & Wu, 2003). Examples of circumstances in which the effectiveness of automated testing tools may be particularly appropriate include (Watkins, 2001):

- testing of applications that have a rapid build and release cycle;
- testing of applications that have to be delivered across diverse platforms;
- testing of applications with complex GUI and/or Client/Server architectures;
- testing of applications that require thorough, rigorous, repeatable testing (e.g., safety critical, business critical, or security critical systems);
- where there is a need to reduce testing timescales and effort; and
- where there is a need to do more thorough testing in the same timescales.

To succeed at automated testing, it is imperative that the selected tool be a good fit with your software technologies (MFC, .NET, Web, etc.) and testing team's technical capabilities (McNaughton, 2005). The tool that is

selected or built must work properly with the selected technologies, and must fit into the project's budget and timeframe allowed for introducing a tool. Evaluation criteria must be established, and the tool's capabilities must be verified according to those criteria prior to purchase (Dustin, 2003). Most experts agree that it's important to test applications for functionality, scalability, and performance at every step of the process: preproduction, production, and postproduction. (Schwartz, 2003).

Bibliography

- Armour, P. (2004, Oct). Not-Defect: The Mature Discipline of Testing. Communications of the ACM, Vol. 47 Issue 10, p15, 4p
- Armour, P. (2005, Jan). The Unconscious Art of Software Testing. Communications of the ACM, Vol. 48 Issue 1, p15, 4p
- Asada, M., & Yan, P. (1998, May) Strengthening software quality assurance Hewitt-Packard Journal, Vol. 49, Issue 2, p. 89-98
- Baer, T. (1997, Dec) Putting APPS to the test. Software Magazine, Supplement Automated Software, Vol. 17 Issue 14, p11, 3p
- Basili, V., & Juristo, N., & Vegas, S., (2002, Dec). What Information is Relevant when Selecting Software Testing Techniques? International Journal of Software Engineering & Knowledge Engineering, Vol. 12 Issue 6, p657, 18p
- Buckler, G. (2005, Feb). Ugly battle brewing over best way to test software. Computing Canada, Vol. 31 Issue 2, p4, 1/2p
- Coffee, P. (2003, Oct) Assuring software quality. eWeek, Vol. 20 Issue 43, p41, 2p
- Copeland, L., (2004). A Practitioner's Guide to Software Test Design. Artech House
- Cram, C., Heldman, W. (2004), IT Project+ Study Guide, Second Edition, Sybex
- Dustin, E., (2003). Effective Software Testing: 50 Specific Ways to Improve your testing. Addison-Wesley
- Feichtner, C., Finkelstein, A., Kappel, G., Retschitzegger, W., Savigni, A., Schwinger, W. (2003, Jan). Ubiquitous Web Application Development -A Framework for Understanding International Journal of Web Engineering and Technology, January 2003.
- Gao, J.Z., Tsao, H.S., Wu, Y. (2003) Testing and Quality Assurance for Component-Based Software, Artech House
- Goodwin, B. (2004, Nov) Testing process must be built in from beginning. Computer Weekly, p14, 1/6p

- Hayes, M., & Whiting, R (2002, Sept). Testing Tools Are Key to Web Services' Success
InformationWeek, no.904, 2 Sept. 2002. p. 56-60. Journal Paper
- Hildreth, S. (2004, Oct). Automating the 40 Monkeys.
Computerworld; Vol. 38 Issue 40, p24, 2p
- Joachim, D (2002, Oct) Testing-1, -2, -3 [automated testing software].
Network Computing, vol.13, no.22, 21 Oct. 2002. p. 81-2. Journal Paper.
- Kan, S. H. (2001). In-process metrics for software testing.
IBM Systems Journal, 2001, Vol. 40 Issue 1, p220, 22p
- Kennedy, R. (2004, Jul). Two Takes on Preproduction Testing. InfoWorld, Vol. 26 Issue 29, p22, 4p
- Laganière, R. & Lethbridge, T., Object Oriented Software Engineering: Practical Software Development using UML and Java, McGraw Hill, 2001
- Langley, N. (2003, Sept) Winrunner automates app testing. Computer Weekly, p62, 1/3p
- Leon, M. (2003, May). Testing Is Key To Success. Computerworld, Vol. 37 Issue 20, p40, 1p
- Li, K., and Wu, M. (2004) Effective Software Test Automation: Developing an Automated Software Testing Tool, Sybex
- Liebmann, L. (2000) Time-To-Market Versus Vendor Lock-In. Network Magazine, Sep2000, Vol. 15 Issue 9, p96, 1p
- Mats, L. (2002, Feb.) The top five software-testing problems and how to avoid them
EDN Vol. 46, Issue 3, p. 107-110
- Mayer, J. (1998, May) The right way to view testing
Software Magazine, Supplement, Vol 18, Issue 7, p. 3 - 7
- McDaid, K, & Wilson, S., Deciding how long to test software
The Statistician (2001) Vol 50, Part 2, p. 117-134
- Pepe, M. (2000, Jul) APPS TESTING SPEEDS TIME TO MARKET. Computer Reseller News, Issue 904, p57, 2p
- Pottage, D. (2001, Nov) Automated Testing Speeds Effective Deployment Of Java Functionality.
Electronic Design, Vol. 49 Issue 23, p65, 3p

QA Labs, (2003). "The Living Creature" – Testing Web Applications
<http://www.qalabs.com/resources/articles/thelivingcreature.pdf>

Rankin, C. (2002). The Software Testing Automation Framework.
IBM Systems Journal, Vol. 41 Issue 1, p126, 14p

Rittinghouse, J. (2004), Managing Software Deliverables: A Software Development Management Methodology, Digital Press

Schwartz, K. (2003, Sept). TOOLS AUTOMATED SOFTWARE TESTING.
InformationWeek, Issue 955, p55, 3p

Shea, B. (2000, July) Software Testing Gets New Respect. InformationWeek, Issue 793, p97, 6p

Stahl, S. (2002, Jan) Poor-Quality Software Needs Zero Tolerance.
InformationWeek, Issue 872, p8, 1/2p

Unknown (2001, Dec). Software Testing and Validation according to FDA Requirements.
Scientific Computing & Instrumentation, Dec2001, Vol. 19 Issue 1, p51, 1p

Unknown, (1997, Jan) ROI enhancers.
Client Server Computing, Vol. 4 Issue 1, Software Quality *Testing* pS13, 1p

Van Tongeren, T (2000, Apr). A Tester's Tale.
Software Magazine; Vol. 20 Issue 2, p46, 3p

Von Halle, B. (2002) Business Rules Applied: Building Better Systems Using the Business Rules Approach, John Wiley & Sons

Watkins, J., (2001) Testing IT: An Off-the-Shelf Software Testing Process.
Cambridge University Press

Wells, T. (2002, Nov). Quality systems give big returns.
Computer Weekly, p26, 1/2p

Wieggers, K. (2003) Software Requirements, Second Edition, Microsoft Press

[This page intentionally left blank.]

Appendix A - Sources Used in Content Analysis

The purpose of this table is to present all of the sources relevant to the scope of the coding rules established in the Method, and elaborated on in the Analysis of Data section. This table represents only the sources that are related to the terms of "Software Quality Assurance", "Automated Testing Software", and "Web Based Applications".

Source	Topic Area
Armour, P. (2004, Oct). Not-Defect: The Mature Discipline of Testing.	Automated testing Software
Asada, M., & Yan, P. (1998, May) Strengthening software quality assurance	Software Quality assurance, automated testing
Baer, T. (1997, Dec) Putting APPS to the test.	Automated testing software, Software Quality Assurance
Coffee, P. (2003, Oct) Assuring software quality.	Software Quality assurance
Dustin, E., (2003). <u>Effective Software Testing: 50 Specific Ways to Improve your testing</u> . Addison-Wesley	Automated testing software, Software quality assurance
Gao, J.Z., Tsao, H.S., Wu, Y. (2003) <u>Testing and Quality Assurance for Component-Based Software</u> , Artech House © 2003	Automated testing software, software quality assurance
Goodwin, B. (2004, Nov) Testing process must be built in from beginning.	Automated testing software
Hayes, M., & and Whiting, R (2002, Sept). Testing Tools Are Key to Web Services' Success	Web based applications, Software Quality Assurance, Automated testing software
Hildreth, S. (2004, Oct). AUTOMATING THE Monkeys. <u>Computerworld</u> ; Vol. 38 Issue 40, p24, 2p	Automated testing software, web based applications
Joachim, D (2002, Oct) Testing-1, -2, -3 [automated testing software].	Automated testing software, Software Quality Assurance, Web based applications
Laganière, R. & Lethbridge, T., <u>Object Oriented Software Engineering: Practical Software Development using UML and Java</u> , McGraw Hill, 2001	Web based applications, Software Quality Assurance

Source	Topic Area
Langley, N. (2003, Sept) Winrunner automates app testing. <u>Computer Weekly</u> , p62, 1/3p	Automated testing software
Li, K., and Wu, M. (2004) <u>Effective Software Test Automation: Developing an Automated Software Testing Tool</u> , Sybex	Automated testing software
Mats, L. (2002, Feb.) The top five software-testing problems and how to avoid them	Testing strategy, Automated testing software
Mayer, J. (1998, May) The right way to view testing	software quality assurance, automated testing software
McNaughton, A (2005, Jan) Avoiding the Pitfalls of Automated testing	software quality assurance, automated testing software
Pepe, M. (2000, Jul) APPS TESTING SPEEDS TIME TO MARKET. <u>Computer Reseller News</u> , Issue 904, p57, 2p	Web based applications, automated testing software
Pottage, D. (2001, Nov) Automated Testing Speeds Effective Deployment Of Java Functionality.	Automated testing software, web based applications
QA Labs, (2003). <u>"The Living Creature" – Testing Web Applications</u>	Web based applications, automated testing software, software quality assurance
Rankin, C. (2002). The Software Testing Automation Framework.	automated testing software
Schwartz, K. (2003, Sept). TOOLS AUTOMATED SOFTWARE TESTING.	Automated testing software, web-based applications
Shea, B. (2000, July) Software Testing Gets New Respect. <u>InformationWeek</u> , Issue 793, p97, 6p	Automated testing software, web-based applications
Van Tongeren, T (2000, Apr). A Tester's Tale.	automated testing software, software quality assurance
Watkins, J., (2001) Testing IT: An Off-the-Shelf Software Testing Process. Cambridge University Press	automated testing software, software quality assurance
Wells, T. (2002, Nov). Quality systems give big returns. <u>Computer Weekly</u> , p26, 1/2p	Software Quality assurance

Appendix B - Contextual Discussions Related to Software Quality Assurance, Automated Testing Software and Web Based Applications

The purpose of this table is to present the resulting data set from the coding of the sources. This table represents all of the contextual data that are related to the terms of "Software Quality Assurance", "Automated Testing Software", and "Web Based Applications".

Source title	Keyword/phrase	Context
Armour, P. (2004, Oct). Not-Defect: The Mature Discipline of Testing.	Automated testing software	Good test software, on the other hand, doesn't just describe the knowledge of testing a system, it executes the knowledge of testing a system
Asada, M., & Yan, P. (1998, May) Strengthening software quality assurance	Software Quality assurance	Increasing time-to-market pressures in recent years have resulted in the deterioration of the quality of software entering the system test phase
	Software Quality assurance	At HP's Kobe Instrument Division, the software quality assurance process was reengineered to ensure that released software is as defect-free as possible.
	Software Quality assurance	Given adequate development time, we are able to include sufficient software quality assurance activities (such as unit test, system test, and so on) to provide high quality software to the marketplace.
	Software Quality assurance	Strong time to market pressure, mainly from consumers and completion has made our development period and the interval between projects shorter. As a result we have recognized two significant problems in our product and process: a deterioration of software quality and an increase in maintenance and enhancement costs
	Software Quality assurance	In recent years (1995 to 1997), software quality has apparently been deteriorating before the system test phase.
	Software Quality assurance	The Software quality assurance department (SWQA) will appropriately revise this plan and improve it based on experience with actual projects.
	Automated testing software	Implementation phases - Automatic test execution. In this phase, SWQA mainly writes test scripts based on function definitions for automatic tests to detect

Source title	Keyword/phrase	Context
		single function defects.
	Automated testing software, Software Quality assurance	When we implement the functions, we immediately execute the automatic tests by using the scripts corresponding to these functions. Thus we confirm the quality of the software as soon as possible.
	Automated testing software	For functions already tested, we reexecute the automatic tests periodically and check for side effects cause by new function implementations.
	Software Quality assurance	As a result of these improvements, we obtain software with no single function defects before the system test phase, thereby keeping the software quality high in spite of the short development period.
	Software Quality assurance	The test scripts are also used in regression testing after shipment to confirm the quality of modified software in the enhancement process
	Software Quality assurance	Through our improved process, we can start the system test phase with high quality code that includes only a few single function defects. Thus we can redefine the testing method to get more efficiency in detecting the remaining defects.
	Automated testing software	Since the automated test is executed periodically during the implementation phase, we can assume that no single function defects remained in Product Y's firmware before system test
	Software Quality assurance	We expect that the quality of Product Z will be high before system test because we test Product Z periodically in the implementation phase and confirm the result before entering system test.
	Software Quality assurance	We analyzed the software development problems of the Hewlett Packard Kobe Instrument division and decided on an improvement process to solve these problems. This improvement process has been applied to two projects: Project Y and Project Z. The results show that we can expect the new process to keep the software quality high with a short development period. The main problems-deteriorating software quality and increasing enhancement cost have been reduced

Source title	Keyword/phrase	Context
Baer, T. (1997, Dec) Putting APPS to the test.	Automated testing software	This article, which first appeared in the October 1997 issue of Software Magazine, highlights how companies are struggling to add automated testing functions into their ERP development projects.
	Software Quality assurance	The company saved even more time and money by streamlining its software testing and quality assurance (QA) process, says Don Wenninger, chief information officer.
	Software Quality assurance	Vixel couldn't afford delays or cost overruns, whatever the cause. So the company accelerated its QA process by focusing solely on end user acceptance testing with the assumption that Oracle's internal software testing had already taken care of applications bugs.
	Automated testing software	Moreover, Vixel eschewed automated testing, further cutting development time and up-front cost.
	Automated testing software	One way to ease the pain is to automated the testing process.
	Automated testing software	Automated testing has proven itself, for example, at Lincoln Insurance Co. in Ft Wayne, Ind., which uses Softbridge's Automated Test Facility on a homegrown C++ customer service application.
	Automated testing software	Since the application is revised quarterly and must be tested each time, it made sense for Lincoln to adopt an automated process.
	Automated testing software	Only Autotester Inc (Dallas) and Mercury Interactive (Sunnydale, Calif) offer testing tools for packaged applications, and early evidence is promising.
	Automated testing software	Still, most ERP customers are not yet aware of the benefits of automated testing.
	Automated testing software	"We didn't use automated tools because we weren't aware of any tools that could help us", said Mary Fonder, IS director for Leeson Electric, an \$142 million manufacturer of electric motors based in Wisconsin
	Automated testing software	For organizations like Leeson, ERP vendors offer rudimentary test automation capabilities.

Source title	Keyword/phrase	Context
	Automated testing software	Still, with the enterprise application market approaching \$4 billion, according to the Meta Group (Stamford, Conn), it appears an awfully fat target for test tool vendors.
	Automated testing software	With or without automated tools, testing an ERP application isn't trivial. At Leeson, implementing Oracle manufacturing and financials took 160 staff months with at least half that spent in function and load testing.
	Automated testing software	As for the actual writing of test cases, tools adapted for ERP packages boast automated record and playback capabilities that reduce the amount of coding or scripting required for developing them. However, these features can be dangerous for the initiated. If the wrong combinations of screens or keystrokes are recorded, the tests could easily career out of control.
	Automated testing software	While automated tools reduce the drudgery of running tests, developing test scripts does take time. "It forces you to do more test case planning. You have to have a testing process and methodology in place", explains Tracy Morgan, testing analyst for T. Rowe Price. But the investment in an automated tool is rewarding when the tests are reused each time the application is modified.
Coffee, P. (2003, Oct) Assuring software quality.	Software Quality assurance	When the subject is software, the concern for today and tomorrow is quality
	Software Quality assurance	For developer rigor in defining and applying quality standards to their work, and for thorough understanding of the more dynamic behavior of code in complex environments such as Microsoft Corp's .net
	Software Quality assurance	The growing concern for software quality swims against a tide of programmer talent that was trained during the early years of the PC to make applications run fast as the principal goal.
Dustin, E., (2003). <u>Effective Software Testing: 50 Specific Ways to Improve your testing.</u> Addison-Wesley	Automated testing software	Automated testing tools can enhance the testing effort - as long as expectations are managed, tool issues are understood, and a tool compatible with the system-engineering environment is selected.

Source title	Keyword/phrase	Context
	Automated testing software	Many types of testing tools are available for use throughout the various development life cycle phases, including the highly touted capture/playback testing tools.
	Automated testing software	In some cases, particularly when investing large amounts of money in automated testing tools, the needs of the entire organization must be considered.
	Automated testing software	The tool that is selected or built must work properly with the selected technologies, and must fit into the project's budget and timeframe allowed for introducing a tool. Evaluation criteria must be established, and the tool's capabilities must be verified according to those criteria prior to purchase.
	Automated testing software	While functional testing tools (also called "captured/playback tools") are much hyped in the testing industry, it is important to be aware of the other types of tools available to support the testing life cycle.
	Automated testing software	Inadequacy of the testing tool is not the only reason why this type of automated testing might be insufficient.
	Automated testing software	Automated testing tools are merely a part of the solution - they aren't a magic answer to the testing problem.
	Automated testing software	Automated testing must be seen as an enhancement to the manual testing process.
	Automated testing software	Along with the idea of automated testing come high expectations. A lot is demanded of technology and automation.
	Automated testing software	Some may incorrectly assume that an automated test tool will immediately reduce the test effort and schedule. While automated testing is of proven value and can produce a return on investment, there isn't always an immediate payback. Sometimes an automation effort might fail, because of unrealistic expectations, incorrect implementation, or selection of the wrong tool.

Source title	Keyword/phrase	Context
	Automated testing software	<p>The following list correct some common misconceptions about automated testing that persist in the software industry, and provides guidelines for avoiding the automated testing euphoria</p> <ul style="list-style-type: none"> · Multiple tools are often required · Test effort does not decrease · Test schedules do not decrease · Automated testing follows the software development life cycle · A somewhat stable application is required · Not all tests should be automated · A test tool will not allow for automation of every possible test combination · The tool's entire cost includes more than it's shelf price · "Out of the box" automation is not efficient. · Training is required · Testing tools can be intrusive · Testing tools can be unpredictable · Automaters may lose sight of the testing goal
	Automated testing software	To be most effective, automated software testing should be treated as a software development project.
	Automated testing software	The automated testing tools currently on the market are not always compatible with all environments or all testing needs.
	Automated testing software	No single automation framework solution is best for all automation requirements.
Gao, J.Z., Tsao, H.S., Wu, Y. (2003) <u>Testing and Quality Assurance for Component-Based Software</u> , Artech House © 2003	Software Quality assurance	In recent years, product and service quality has become a principal means for many corporations to distinguish themselves from their competitors. IEEE defines software quality as "the degree to which software meets customer or user needs or expectations"
	Automated testing software	In past decades, people have learned that software testing and maintenance usually costs 40% to 60% of a software project budget [1]. Due to the increase of business competition in the real world, many software workshops are trying their best to shorten the software development life cycle. Hence, software test managers are looking for the solutions to reduce testing costs and time in a test process. In the past decade, software test automation has been approved to be one

Source title	Keyword/phrase	Context
		of the effective solutions.
	Automated testing software, Software Quality assurance	<p>Software test automation is one important and effective means to reduce manual testing effort and cost and to speed up and improve the quality of a software test process. In our opinion, successful test automation projects and efforts change a manual test process in the following areas:</p> <p>Automate testing operations and reduce manual retest efforts.</p> <p>Enforce the establishment of software testing and quality assurance standards, and enhance the quality of tests.</p> <p>Replace ad hoc testing practice with systematic testing solutions and well-defined test criteria.</p>
Goodwin, B. (2004, Nov) Testing process must be built in from beginning.	Automated testing software	Automated testing tools can be helpful, but they need to be used with care, said Daniel, Dresner, security consultant at NCC
	Automated testing software	"You have to realize that whatever changes you make to the system can affect the test tools. If you set up a test sequence based on a certain design of software and you change your software, the testing sequence is not valid", he said
Hayes, M., & and Whiting, R (2002, Sept). Testing Tools Are Key to Web Services' Success	Automated testing software	What can companies do to ensure that the web services they build will operate flawlessly and stand up to the rigors of E-business? One answer is software that tests applications performance and functionality
	Automated testing software	While many vendors are still only planning to develop web-services testing tools, companies such as Compuware, Empirix, Mercury Interactive, Parasoft, Rational Software, and Segue software are already putting down stakes in this new territory.
	Automated testing software	Some early adopters of web services are finding value in this new breed of testing tools.

Source title	Keyword/phrase	Context
	Automated testing software	Financial services firm ABN Amro Bank N.V. is using Parasoft's SOAPtest, which became widely available in April, to test adapter software it developed called Wolf (Web Objects for Legacy Functions).
	Automated testing software, Software Quality assurance	The Parasoft SOAPtest tool that ABN Amro is using measured three main areas of quality testing: Functionality (how well the program works), load (how it performs under strain), and regression (whether any code changes result in problems).
	Automated testing software	"SOAPtest is a very good and very fundamental tool," says Paul Raj, lead quality assessment engineer at ABN Amro. "It's allowing you to intercept the Soap characters as they come along and examine them to make sure they're what you expect they'll be."
	Automated testing software, software quality assurance	Many companies are buying software that tests for quality in code and applications, but it's use isn't ubiquitous.
	Automated testing software	An informationWeek Research survey of 800 business-technology managers in April found that 60% of companies are using commercial testing software
	Automated testing software, Software Quality assurance	Web services will require higher-quality software, and the only way you can automate testing is through the tools that are offered
	Software Quality assurance	Because Web services are such a new technology, testers also lack standards such as performance benchmarks for measuring things like quality of service levels, Narsu says.
	Automated testing software, web based applications	Mercury Interactive, a nearly \$400 million a year company and the largest of the software testing vendors, is building Web-services support into its product line, rather than developing new products that target the area. In February, it updated LoadRunner, it's software for testing systems under heavy use, and Astra LoadTest, for testing scalability and performance, to support web services applications built with Microsoft Visual Studio .Net, an XML based development environment

Source title	Keyword/phrase	Context
	Automated testing software	Mercury Interactive says a new product it released in July, called ProTune, which lets companies tune applications for optimal performance before deploying them, was built with support for the Web services protocols and open interfaces.
	Automated testing software	Mercury Interactive's tools play an important role in measuring the performance and functionality of that Soap middle tier.
Hildreth, S. (2004, Oct). Automated the 40 Monkeys. <u>Computerworld</u> ; Vol. 38 Issue 40, p24, 2p	Automated testing software	"If 50 people find problems, which I fix and then want to do another test...well, it just isn't easy to get everyone back again", says Dave Wollin, managing group director of IT at AXA Financial Inc. The New York based finance and insurance company used Mercury Interactive Corp's testing tool before deploying a Web application to field agents
	Automated testing software	The desire for repeatability and accuracy is one reason organizations are moving to automate testing, both to uncover bugs and ensure that they meet performance standards
	Automated testing software	Melissa Webber, research director at IDC in Framingham, Mass., says worldwide sales of automated testing tools rose by 8.5% in 2003 and are expected to grow at an annual compound rate of 9.3% through 2007. "This sector has recovered faster than the overall IT market", she says.
	Automated testing software	The increasingly integrated nature of IT systems is another factor pushing companies towards automated testing. Human testers can't always accurately recreate the flow of transactions between applications. That's why Varian Medical Systems Inc., a Palo Alto, Calif. based maker of medical devices, opted to purchase an automated functional testing tool, SilkTest, from Segue Software Inc. last year
	Automated testing software	"Our applications pass data like treatment and patient information, so there is a real time flow between them. We could only test the whole system effectively with automation", explains Ashish Katrekar, quality assurance manager at Varian.

Source title	Keyword/phrase	Context
	Automated testing software	Also, manual testing added as much as six months to Varian's development cycle, seriously lengthening the time to market for its products. "Our test suites had grown so big that we couldn't run them quickly. We needed automation to achieve a quicker turnover of products." says Katreakar.
	Automated testing software	Vendors such as Mercury, Segue Software, Computerware Corp., IBM's Rational unit, RadView Software Ltd., and Empirix Inc. offer products for testing client/server and Web based applications, as well as mainframe and other legacy systems.
	Automated testing software	"If you are using an automated test tool and your testers don't lay a hand on the system, your users will be turning up bugs you had no idea were there - because you never thought to test for them", observes James Lyndsay, principal consultant at Workroom productions, a software testing and development company in London.
	Automated testing software	Human testers may also be appropriate for tests that won't be repeated. Automated tools take time to set up and learn, so one time tests are often done faster manually. But for repeated use, automation is usually quicker and more accurate.
	Automated testing software	"There are always new rules and requirements, so this is not an application that we'll finish and never touch again," explains Idriss Mekrez, senior software quality architect for NASA's Peer Review services. "Automated testing allows us to define the basic functionality the system needs to deliver and test those processes automatically".
	Automated testing software	For mission-critical applications, automated testing is usually well worth the effort, given the consequences of failure. "The applications that most need testing are those with the biggest impact on the business", says Dan Koloski, product manager at Empirix. "If an application is customer facing and revenue generating, it is ripe for substantial testing".

Source title	Keyword/phrase	Context
Joachim, D (2002, Oct) Testing-1, -2, -3 [automated testing software].	Automated testing software, Software Quality assurance	"We were developing software faster than the QA team could test it", says Phil Petersen, development team lead. "Getting the code qualified was the bottleneck". To break the logjam, Life Time is combining software development and testing into what Zempei calls a "single, seamless process," using automated testing software called WinRunner, from Mercury Interactive. The change has done more than simply help the three person QA team catch up to the six person application development staff - it has also shortened the development cycle times and improved the quality of the code at the development stage.
	Automated testing software	Mercury's Winrunner is used to perform regression testing, which ensures that an upgrade to one piece of software, such as a web server, doesn't break an application built to run on that server. Winrunner also automates the testing of common routines
Laganière, R. & Lethbridge, T., <u>Object Oriented Software Engineering: Practical Software Development using UML and Java</u> , McGraw Hill, 2001		
Langley, N. (2003, Sept) Winrunner automates app testing. <u>Computer Weekly</u> , p62, 1/3p	Automated testing software	Winrunner is an automated tool for testing enterprise applications before deployment by making sure they will not fall over, and after they have gone live by ensuring they perform acceptably.
	Automated testing software	Winrunner captures, verifies, and replays user interactions automatically, enabling developers to build test "scripts" which when run against an application, identify defects.
	Automated testing software	Automated testing removes much of the complexity of manual testing by emulating large user populations; coordinating the interactions of different applications, databases, browsers, clients, servers and networks; and building test suites that can be adapted incrementally as systems evolve.

Source title	Keyword/phrase	Context
	Automated testing software	Mercury Interactive says 75% of Fortune 500 companies use its tools.
	Automated testing software	But while tools such as this help with scanning large applications, they are not a replacement for the skills of a trained software tester.
Li, K., and Wu, M. (2004) <u>Effective Software Test Automation: Developing an Automated Software Testing Tool</u> , Sybex	Software Quality Assurance	<p>To ensure software quality, organizations encourage software developers and testers to achieve objectives such as these:</p> <ul style="list-style-type: none"> Locating the source of defects faster and more precisely Detecting bugs earlier in the software development life cycle Removing more defects before the product is released
	Automated Testing Software	<p>Improved testing tools can reduce the cost of software development and increase the quality of software. An automated testing tool must have the following characteristics:</p> <ul style="list-style-type: none"> Accurate functionality, reliability, interoperability, and compliance An interface that is user friendly and easy to learn and operate Enhanced fault tolerance and automatic error recoverability Efficient algorithm for time and resource management Stable and mature final products that can be maintained and upgraded Easy portability with regard to installation, uninstallation, adaptability, and security
Mats, L. (2002, Feb.) The top five software-testing problems and how to avoid them	Automated testing software	Manually testing a complex system with real time requirements is, at best, unreliable, and at worst, impossible. Fortunately, significant research and world standardization has occurred in the last 10 years that makes possible reliable automated testing of this type of system.

Source title	Keyword/phrase	Context
	Automated testing software	The ISO (International Standards Organization) has arrived at a standard (9646) on formalized and automated testing of communicating and real time software, called the TTCN (Tree and Tabular Combined Notation)
	Automated testing software	Make sure from the start that you can test your design using automated methods.
	Automated testing software	Given the complexity of current and anticipated software and communications systems, you should expect software testing to become even more complicated. Consequently, even more potent tools and methodologies will emerge over time.
	Automated testing software	Manual testing is becoming a less viable alternative, and integration with the overall design processes and tools will prove necessary to keep pace in testing these complex current and future systems.
Mayer, J. (1998, May) The right way to view testing	Software Quality assurance	Worse, software quality is rarely a selling point. It's usually viewed simply as an expense. As a result, code developers and other staff often assume that the job is not only unimportant, but actually an impediment to time-to-market goals.
	Software Quality assurance	Although they're often mistakenly referred to as one and the same, software quality assurance and testing play critical roles in the successful implementation of virtually any application.
	Automated testing software, Software Quality assurance	At Digital Equipment Corp. in Maynard, Mass., for instance, software QA is helping the company meet critical implementation guidelines for a new worldwide SAP R/3 project. In fact, using R/3 testing tools from Autotester Inc., Digital's QA group has helped cut costs by over \$250,000.
	Software Quality assurance	According to a recent survey of IS professionals conducted by The Sentry Group, Westborough, Mass (publisher of this magazine), 62% of all organizations don't even have a formal QA or test group, and a large majority place a much higher priority on meeting schedule deadlines than producing high-quality software.

Source title	Keyword/phrase	Context
	Automated testing software, Software Quality assurance	So how can test and QA managers make a business case for their functions? And can they do so without pushing out product delivery dates? The key, agree IS managers and test tool vendors, is to sell QA and test and integral parts of the overall development process.
	Software Quality assurance	Building in quality via rules and standards during the code development process allows teams to minimize the number of defaults downstream and allows test teams to statistically check the quality of code and shorten the test process.
	Software Quality assurance	To get a better handle on the entire development process from product and feature inception to release, MedicalLogic began a process that saw software QA getting involved earlier and earlier in development.
	Automated testing software	Building confidence in an application and reducing risk without delaying product release remain the greatest challenges that QA and test managers face. Automated development tools are helping meet time-to-market deadlines. The QA group at Digital, for instance, used AutoTester's AutoController to simulate 2,000 concurrent users performing tasks such as logging on and creating production orders on its new R/3 implementation.
	Automated testing software	The latest breeds of tools fill another critical requirement in any integrated development and test environment - information. As management is caught between the increasing complexity of applications and shrinking product development cycles, comprehensive management information has become essential to understanding what's happening during testing and whether a team has tested its application enough.
	Software Quality assurance	At the start of testing, Benson's group makes assumptions using historical data on the number of defects they can expect to discover during testing, and on their ability to repair problems given their staffing levels. Then using a simple linear model, the group calculates what kind of quality level they can achieve by a given date.

Source title	Keyword/phrase	Context
	Automated testing software	Using an automated test management system called TestExpert from Silicon Valley Networks Inc., Benson's team can automatically identify which tests are of the highest priority.
	Automated testing software	"Say you're short on testing time because you have a market window you want to make, yet you have 4,000 tests to perform, " says Benson. "You want to make sure you test all those functions that are critical to the customer. Automated test management makes it much simpler to pick the cases that fit those needs"
	Software Quality assurance	Secondly, the QA manager has to argue that quality assurance extends beyond testing the code to trying to put in place an effective development process, because if an organization is simply testing its applications on the back end, it can't input quality into a product.
McNaughton, A (2005, Jan) Avoiding the Pitfalls of Automated testing	Automated testing software, Software Quality assurance	Quality assurance engineers are being asked to take on more complex projects than ever before. Application complexity has increased to the point that formerly simple test plans now encompass hundreds, if not thousands, of test cases that must be executed under stringent deadlines. These demands have resulted in quality assurance teams seeking higher efficiencies than traditional manual testing provides. Much of the focus of this search has been on automated testing tools
	Automated testing software, Software Quality assurance	It is commonly thought that automated testing is the panacea for all quality assurance woes. But is that always the case?
	Automated testing software, Software Quality assurance	While a well executed automated testing strategy can measurably improve productivity, in reality there are many pitfalls waiting to ensnare the unwary. Quality assurance teams need to carefully analyze the path before them. By choosing the right tool for their needs and using it wisely, they can reap the much heralded benefits of automated software testing
	Automated testing software	To succeed at automated testing, it is imperative that the selected tool be a good fit with your software technologies

Source title	Keyword/phrase	Context
		(MFC, .NET, Web, etc.) and your testing team's technical capabilities.
	Automated testing software	Automated testing projects must be carefully planned to ensure success.
	Automated testing software	Determining which parts of the application are right for automated testing and which parts should continue to be tested manually is straightforward when the following guidelines are heeded <ul style="list-style-type: none"> · Identify components of the application that require large amounts of data entry. · Identify areas of the application that require repetitive testing with different data values for each iteration. · Identify portions of the application that require data comparison or validation. · Identify which parts of the application deal with complex business rules. · Avoid automating areas of the application that change constantly.
	Automated testing software	Automated testing holds great promise. The tyranny of repetitiveness can be broken when the right tool is used intelligently. To reach this goal it is imperative that ease of use be seriously considered when selecting an automated testing tool
	Automated testing software	This simplicity lowers cost and enhances productivity by requiring less training and fewer resources than language based tools
Pepe, M. (2000, Jul) APPS TESTING SPEEDS TIME TO MARKET. Computer Reseller News, Issue 904, p57, 2p	Automated testing software	Given the complexity of web and e-commerce software, the onus falls on individual developers to unit-test, or test their own application components, say analysts. A number of products aim to simplify this process via automation.
	Automated testing software	Other players too, including Rational Software, Mercury Interactive, and RSW software, have worked successfully in the past six months to automate testing and streamline the tools used by development teams, say analysts.
Pottage, D. (2001, Nov) Automated Testing Speeds Effective Deployment Of Java Functionality.	Automated testing software	This article discusses the benefits and challenges of test automation, plus its use in effectively and quickly deploying Java functionality in these highly diverse devices.
	Automated testing	Some benefits of test automation include:

Source title	Keyword/phrase	Context
	software	<ul style="list-style-type: none"> · Faster than manual testing · Repeatability to identify regressions immediately · Insulation from human factors, such as boredom or carelessness · Improved audit trail to identify bugs · Shorter residence time of bugs in the code, reducing the manual effort of development and test engineers · Ability to test code in live development
	Automated testing software	The main challenge of automation is the difficulty of implementing it.
	Automated testing software	Some tests, particularly those involving devices that feature a graphical user interface, are hard to automate.
	Automated testing software	Automation generally takes longer to setup and requires more skilled software engineers than a manual test
QA Labs, (2003). "The Living Creature" – Testing Web Applications	Software Quality assurance	Remember, many different companies define quality in many ways, but to the end user, quality always means, "Am I satisfied?"
	Software Quality assurance	If we define quality as perceived value to the user over cost to the user, then web applications, which have no real cost to the user, should move towards infinitely greater quality.
	Software Quality assurance	Decreasing quality of software, for both desktop applications and web applications, is something that has been occurring for quite some time now, and the sad truth is that end users have been tolerating (if not encouraging) this trend.
	Software Quality assurance	As the internet becomes swamped with dot.coms competing against each other, the limited attention span of users will reward only those sites that do not disappoint the user. This is the key argument for increasing pressure towards better testing of web applications: with traditional software a user has spent a certain amount of money and hence feels motivated to get the best utility from his or her investment. We believe that quality and functionality of web sites will ultimately be a major factor in the inevitable shake-out, which will reduce the number of redundant dot.coms.

Source title	Keyword/phrase	Context
	Software Quality assurance	Some key quality factors that can be related to web applications include <ul style="list-style-type: none"> · Reliability · Recoverability · Security · Usability · Performance
Rankin, C. (2002). The Software Testing Automation Framework.	Automated testing software	Automation of execution related activities offers another potential source of savings in the testing process.
	Automated testing software	For purposes of this paper, a test is a program executed to validate the behavior of another program. Automation refers to the removal of human interaction with a process and placing it under machine or program control. In our case, the process in question was software testing
Schwartz, K. (2003, Sept). TOOLS AUTOMATED SOFTWARE TESTING.	Automated testing software	Back in the 1990's, when IT budgets were particularly tight, testing tools were often the first victims, deemed unnecessary in the face of more pressing concerns. Often, IT managers dispensed with the use of automated testing tools when developers were running late, figuring that manual testing would suffice.
	Automated testing software	Vendors catering to the software testing market are incorporating more advance features, and integrating various types of testing capabilities; to create more coordinated suites of testing tools. Today's tools are more efficient, accurate, and thorough. They're also easier to use and can test a greater variety of functions.
	Automated testing software, Software Quality assurance	Witness financial services company H & R Block Inc. Although quality assurance has been part of a structured approach to software testing for the company for the past eight years, "it's when the processes began to mature and we began to learn more about quality assurance and the types of tools that were available that we became more interested and willing to invest in those types of tools" says Mike Deloney, director of quality assurance. Today H & R Block uses tools from a variety of software testing companies, including Compuware Corp. and Empirix

Source title	Keyword/phrase	Context
		Inc.
	Automated testing software, Software Quality assurance	<p>Most experts agree it's important to test applications for functionality, scalability, and performance at every step of the process: preproduction, production, and postproduction. For preproduction, or development, there are three types of tools</p> <ul style="list-style-type: none"> · Developer-oriented tools. Also called component or unit testing, these tools address memory leaks and other early performance problems. They're used to test specific pieces of the developer's application code independent of other units of code within the same app. They're especially useful in testing large applications where different developers are working on different parts of the app. · Functional testing. These tools, usually used by a company's quality assurance department, test that the code being developed is acting as expected. All tools in this category have scripting capabilities, letting developers modify what they've captured to test additional items. One capability of a functional testing tool is capture/replay, which runs sample tests against a working version of a program, capturing the activity it generates. During the playback phase, developers can see whether they're getting the expected results. Another subset of functional testing tools is regression testing, which tests whether a specific function continues normally after other changes have been made to the code · Load testing. Also called stress or performance testing, these tools test what happens to the code when multiple users access the application simultaneously. Load testing is more critical in a Web environment, because it can be difficult to predict the volatility of the load change.

Source title	Keyword/phrase	Context
	Automated testing software	<p>Testing after the application goes live requires a different set of tools. These include:</p> <ul style="list-style-type: none"> · Performance monitoring and management. These tools rely on a variety of metrics gathered by monitoring the use of the application. Tools in this category will let a tester run a script to determine the type of performance the app is providing at different points during the day. If performance falls below a defined threshold, the system can be set up to notify a developer, who can then find the problem and fix it. · Test management. This type of tool manages testing across the board, letting developers and quality assurance managers track tests that are being run on various applications, trace those tests back to the original requirements, and determine what errors were found during those tests.
	Automated testing software	Testing tools have matured in the past few years, and some of the most significant advances have occurred in functionality, ease of use, and support for development environments.
	Automated testing software	Today's tools are more scalable and easier to maintain than previous versions, making them easier to use and understand for business analysts who have the most domain knowledge of the application.
	Automated testing software	Years ago, testing tools often supported just one or two obscure or proprietary languages, making them difficult to use or even irrelevant in development environments with a variety of languages. Today, more testing tools support standard programming languages such as the Java 2 Enterprise Edition and Microsoft Visual Basic.
	Automated testing software	Today, testing tools offer more point and click script building, opening this important testing method to a variety of users. These tools also offer greater customization, letting testers use scripting languages to write their own scripts. "Script maintenance is the main challenge to getting people to use testing tools consistently", says Billie Shea,

Source title	Keyword/phrase	Context
		director of research at Newport Group
	Automated testing software	Today's testing tools offer better traceability notification. The process of trying test cases to requirements and tracing defects back to those requirements has improved significantly.
	Automated testing software	And testing tools today offer more comprehensive suites. Originally, testing vendors developed point solutions - performance solutions for load testing, functional testing solutions to make sure the code was acting as expected, and component testing to address early performance problems. Although standalone tools still exist, and can be effective for small or specialized testing, more vendors are developing testing suites.
	Automated testing software	"If you have to write tests and scripts for the functional tools, you can carry them forward and use some of the same ones in load testing. The load testing tools can find the problem and call in a developer oriented tool to drill down and help fix it" says Dick Heiman, research director for application development and deployment at research firm IDC
	Automated testing software	The future of automated testing lies in several areas: Web services, predictive analysis, and hosted and outsourcing scenarios.
	Automated testing software	Many developers are calling for a new category of software testing tool, one that can predict when something is going to break. Dubbed predictive analysis, this type of tool might, for example, help determine whether an application will use network resources appropriately within a given environment.
	Automated testing software	More developers are using automated software testing tools than ever before, and for good reason: The increasing complexity of software development demands it. It pays to understand the capabilities available - now and in the future - in this critical area.
Shea, B. (2000, July) Software Testing Gets New Respect. <u>InformationWeek, Issue 793</u> , p97, 6p	Automated testing software	Long ensconced in backroom IT offices as the Rodney Dangerfield of the application development process, software testing - specifically of the automated type - is gaining newfound respect.

Source title	Keyword/phrase	Context
	Automated testing software	Companies intent on transacting revenue generating business or offering enhanced customer services online are increasingly turning to automated testing solutions to gain confidence in their IT systems, fully understand how applications will behave under real world conditions, uncover and rectify issues, and systematically manage growth
	Automated testing software	The worldwide market for automated software quality tools, including mainframe and distributed environments, reached \$931 million in 1999, a 23.6% increase over 1998 figures, according to recently published research from International Data Corp. And the market is slated to grow to more than \$2.6 billion by 2004
	Automated testing software, Software Quality assurance	Dick Heiman, IDC's research director and analyst, says the widespread growth in the adoption of automated testing solutions is being fueled by the steps that businesses are taking to leverage the Web. "The old paradigm of forgoing a structured quality initiative in exchange for faster deployment - with plans to address quality issues with application updates - doesn't fly in the Internet economy"
	Automated testing software	Business acceptance of automated testing as a mainline business practice hasn't come quickly or easily. For traditional client server environments, it's been more common for testing efforts to be shortchanged in exchange for more development time or faster deployments.
	Automated testing software	Internal conflicts that result in poor communication between developers and testers have also contributed to weakened efforts, along with a loss of focus on the fact that both facets of IT must work together to achieve the end result. Too often, automated tools have been shelved due to inadequate test process support, lukewarm business management support and staff turnover

Source title	Keyword/phrase	Context
	Automated testing software	However, in the ever-widening Internet economy, much is changing. According to Brian Chase, quality assurance manager for Thomson & Thomson, a provider of legal research, "Today, automated testing tools are viewed as a necessary purchase and we have annual budget allocations for them". Initially approved for purchase on the levels of the company's Y2K testing efforts, Chase says "we've proven that automated testing solutions work to help us deliver our web-based product to market sooner, with more accuracy and less user-found errors. In turn, business support continues to strengthen".
	Automated testing software	The equation is simple: Application performance and transaction precision equate to the efficient business services that lead to customer satisfaction, which ultimately boils down to revenue. So it goes that IT managers and business executives are speaking the same language - bottom line revenue - and, therefore, the business investments in automated testing solutions are more easily justified and understood.
	Automated testing software	With the unique experience of obtaining business management buy in for automated tools in closed client-server environments and open, Web enabled enterprises, Wilkerson says the growing volume of business via the Web has increased business support for test automation. "There's more education about automated testing at the business level, and a greater awareness about the consequences that result when things don't work", he says.
	Automated testing software	CIO Nathan Harper of Guild.com Inc., an online source for purchasing contemporary art, also recognizes a major upward shift in the business value that automated testing has gained as a direct result of doing business online. Harper received no business resistance to his request to purchase automated tools and consulting services for the E-Commerce startup. "The recognition of the value that automated testing brings to the business has changed enormously"

Source title	Keyword/phrase	Context
		says Harper.
	Automated testing software	Web development is embracing more structured processes, and automated testing is an integral part of those processes.
	Automated testing software	"As growth continues to accelerate, we're focused on reconfiguring software and rearchitecting the Web infrastructure", says Chase. "As a result of a better thought out Web architecture and testing processes that included structured and repeatable functional regression and load tests, we're more confident in our system"
	Automated testing software	The "gut instinct" testing mentality that previously existed for determining readiness has been replaced with quantifiable facts about how the application will perform under real world conditions. Chase reports that communication with the development team has improved as a result of leveraging automated tools. "Now when we uncover problems, we can back up our findings with the information our developers need to resolve the issues"
	Automated testing software	Many an IT manager has pitched a case for purchasing automated testing solutions and been challenged to prove the business value. The cost justification: How long before the investment outweighs the cost? After all, the foray into automating the testing process is expensive. Automated tools are costly, as are the qualified staff and hardware resources necessary to operate and maintain them.
	Automated testing software	According to Harper, Guild.com spent \$100,000 on testing tools in 1999 and has allocated another \$60,000 this year. This doesn't include the multiple quality assurance hires, required hardware, or the outsourced testing consultant the company retains. This is why justifying the purchase and a commitment to automated testing has traditionally been uphill battles. However, the cost of not testing applications and eradicating terrors carries an even higher price tag, a fact that easing the budgeting pain

Source title	Keyword/phrase	Context
	Automated testing software	Our beginning.com CEO Michael Budowski has changed his views on the importance of automated testing. "When our IT team came to us with the suggestion of purchasing an automated testing solution via a hosted testing service, our executive team didn't give it much consideration because we were looking at it in terms of dollars and cents. However when we stepped back and evaluated our seven figure investment into the development of our site, and the fact that we were set to advertise during the Super Bowl, we decided to approve the purchase". In the end it was a good business decision because the testing did find errors. After making the appropriate adjustments, Our beginning.com was able to handle the 1600% surge of visitors that flooded its site after the Super Bowl ad ran.
	Automated testing software, Software Quality assurance	What tends to most easily convince businesses of the need to establish formal automated test processes is a bad experience with an application providing poor service, or one that fails altogether. The upside of that experience is that application quality becomes a bigger issue and gains the business support needed to survive. The downside is that adopting automated tools in response to a crisis can seem to worsen problems if it's assumed that having the tools will immediately equate to better application quality.
	Automated testing software	According to Steve Marshall, CEO of T-Plan Ltd., a provider of test process management tools, "typical client consulting engagements require 75% of the training to focus on the test process, with the remainder spent on learning how and where to leverage automated tools"
	Automated testing software, Software Quality assurance	Automating the testing process is a formidable challenge and it's an ongoing process, forging a continuous path of quality assessment and application improvement

Source title	Keyword/phrase	Context
	Automated testing software	In response to growing business acceptance of automated testing solutions, vendor are strengthening their tool suites and emerging with flexible testing service options. That's what web services vendors such as Compuware, Cyrano, Mercury Interactive, Rational Software, and Segue Software are doing.
	Automated testing software	In addition to the traditional vendors responsible for turning automated testing into a distinct discipline, there exists a new breed of vendors that are providing automated testing solutions built exclusively for the web.
	Automated testing software	For all these vendors, the web testing space is realizing the highest customer demand and, thus, potential for growth.
	Software Quality assurance	As more companies work to develop and deploy applications for the Internet economy, by default there's a larger audience faced with the challenges of maximizing quality. Constant time to market pressure is only one issue. For many startup companies, financial constraints limit the ability to purchase tools, hardware, and hire people.
	Automated testing software	Given the undeniably complex array of technologies and the unpredictability of users' loads inherent in the Web-enabled enterprise, automated testing practices will continue to gain business acceptance for companies participating in the Internet economy.
	Automated testing software, Software Quality assurance	The amount of test automation that's chosen to support the quality of any IT system comes down to finding the right balance within individual companies, but it's clear that there are more choices available today to help the testing effort along.
	Automated testing software	Rapid release cycles and continuous changes make test automation a more practical and reliable way to ensure quality IT systems
	Automated testing software, Software Quality assurance	Perhaps the time has arrived for businesses to tear a page out of Henry Ford's lesson book and leverage test automation in a manner that works to make quality "Job One" for the web enabled enterprise

Source title	Keyword/phrase	Context
Van Tongeren, T (2000, Apr). A Tester's Tale.	Automated testing software, Software Quality assurance	Automatic test case generation lowers costs by increasing quality and productivity, and helps developers find errors earlier in the life cycle.
	Automated testing software	With deadlines tightening and workloads increasing, software engineers are constantly looking for ways to improve the process of software development. They are streamlining their customers processes, so why not maximize their own efficiency. One way to do so is with automatic test case generation
	Automated testing software	Some people may be tempted to automate the entire process from requirements to implementation, but from my experiences in test automation, a human will usually find defects the system doesn't.
Watkins, J., (2001) Testing IT: An Off-the-Shelf Software Testing Process. Cambridge University Press	Automated testing software	The majority of test automation tools [1] allow re-usable tests to be created by recording the behavior of the AUT under its typical operation, then subsequently replaying those tests (against a later build or release of the software) to compare the observed behavior against the expected behavior (i.e., the originally recorded behavior). Where the observed behavior differs from the expected behavior, the tool logs this event and allows the tester to analyze how and why the differences have occurred
	Automated testing software	A significant benefit of such tools is their ability to operate unattended (e.g., to run overnight or over the weekend), providing significant gains in productivity for the testing task. Automated tools can also provide additional confidence in the AUT by allowing more tests to be executed in the same time as that taken for an equivalent manual test.

Source title	Keyword/phrase	Context
	Automated testing software	<p>Examples of where automated testing tools may be particularly appropriate include:</p> <p>Testing of applications that have a rapid build and release cycle</p> <p>Testing of applications that have to be delivered across diverse platforms</p> <p>Testing of applications with complex GUI and/or Client/Server architectures</p> <p>Testing of applications that require thorough, rigorous, repeatable testing (e.g., safety critical, business critical, or security critical systems)</p> <p>Where there is a need to reduce testing timescales and effort</p> <p>Where there is a need to do more thorough testing in the same timescales.</p>
	Automated testing software	<p>Finally, although it is possible to gain rapid benefits from the introduction of automated tools (see Reference 17, for example), typically the benefits that such tools bring will not be realized during their initial use, since the tools will require integration into the existing testing process and there will be learning curve issues to be overcome. However, it is important that organizations persevere with their use; otherwise such tools are likely to quickly become "Shelfware," and the initial investment in the tool and its use will be lost.</p>
Wells, T. (2002, Nov). Quality systems give big returns. <u>Computer Weekly</u> , p26, 1/2p	Software Quality assurance	Testing software and systems for Quality will play a large part in the future IT landscape
	Software Quality assurance	A staggering amount of capital expenditure today is spent on software and systems. It was only a matter of time before difficult questions were asked about the quality of those systems and ultimately the return on investment