Automated Web Testing Toolkit

Expert Methods for Testing and Managing Web Applications

Diane Stottlemyer

Wiley Computer Publishing

John Wiley & Sons, Inc.

NEW YORK • CHICHESTER • WEINHEIM • BRISBANE • SINGAPORE • TORONTO

Publisher: Robert Ipsen
Editor: Cary Sullivan
Assistant Editor: Christina Berry
Managing Editor: Marnie Wielage
Associate New Media Editor: Brian Snapp
Text Design & Composition: Carlisle Communications, Ltd.

Published by John Wiley & Sons, Inc.

Published simultaneously in Canada.

Library of Congress Cataloging-in-Publication Data:

Stottlemyer, Diane.
Automated web testing toolkit : expert methods for testing and managing Web applications / Diane Stottlemyer,
p. cm.
Includes index.
ISBN 0-471-41435-2 (pbk. : alk. paper)
1. Computer software—Testing. 2. World Wide Web. I. Title.
QA76.76.T48.S76 2001
005.2'76—dc21
Printed in the United States of America. 2001026006
10 9 8 7 6 5 4 3 2 1

This book is dedicated to Donna for working with me and encouraging me to put my thoughts in print.

This page intentionally left blank.

Contents

Acknowledgments

I would like to thank:

Sierra Roberts (Parasoft Software) for providing information on JTest

Noelle Beaudin (Cyrano) for providing information on Cyrano's line of free products

Ann Hewitt (Empirix) for providing information on eTest

Jenny Jones (Segue) for providing information on Segue Software

Stefan Asbock (Segue) for providing additional information on the CD

Donna Bridgham (Sr. Programmer) for helping to check content as the book was being written

Brendan O'Connell (Compuware) for providing testimony and solutions from Compuware

Microsoft Corporation for having detailed documentation available online

Carnegie Mellon for the Cert Web site that provided security information

I would like to thank Cary Sullivan, Christina Berry, and Marnie Wielage at John Wiley & Sons for all of their hard work, patience, and support for my first book.

I would also like to thank anyone else who was involved in this endeavor and to all the testers who make the quality of what you see better.

This page intentionally left blank.

About the Author

Diane Stottlemyer is a Certified Software Test Engineer. She was involved in several Y2K projects for Fortune 500 companies. She also has a programming background and has taken part in several Web testing projects.

Diane is a graduate of Indiana University and has received her Masters Degree in Computer Science. She has also completed all of her coursework for her doctorate in Computer Science and is completing her dissertation. She is presently teaching for ElementK, CalCampus, Connected University, and Learning Tree. Diane is also a faculty member at Capella University where she teaches four courses: Presentation Layer: Client Side Programming, System Assurance Quality and Testing, Enterprise Application Testing, and Project Estimation and Budgeting. Diane also is a faculty member at Franklin University where she teaches Database Management Systems. She will also be teaching at Mary Baldwin College in the fall.

In her spare time, Diane enjoys looking through technical books and magazines, and makes time to read a good fantasy book. Diane is presently gathering data for her second book.

This page intentionally left blank.

Introduction

This book will address the recent changes in the field of Web development as they apply to Web testing. It will help ensure that developers, Webmasters, and testers are not only able to build and test applications quickly, but to test for full functionality of the Web site. Developers and testers are responsible for code changes, enhancements to the Web site, and the process of regression testing. As these changes occur it is necessary to be able to test the Web site repetitively. This book will address how testing can be implemented and handled to ensure that when code modifications are made to the Web application, a systematic approach to testing is available.

The field of testing is a somewhat overlooked aspect of the entire software and Web site development process. Testing is an essential phase of the software development life cycle as well as Web site life cycle development. This book is a valuable resource for developers, software managers, and testers because it addresses Web design, Web architecture, Web servers, ISP providers, Web testing, and other related topics essential to understanding the testing process.

The unique feature of this book is not only the emphasis on Web software testing, but also the basics of testing and management processes. Since the current trend is moving more toward business on the Internet, this book will be an asset to individuals that would like to have guidance in the area of testing—more specifically Web testing.

Overview of the Book and Technology

The focus of this book is to provide you with the necessary tools to design, test, and implement your site. It is a must read if you need to understand what kinds of tools are available, what the tools can do, and how to get the pertinent information you need to make an educated decision that will be best for your Web site.

The aim of this book is to inform testers, potential testers, project managers, and others about what is available for testing Web sites. This book is structured

to take you from the earliest steps of testing through completing the testing process. You will be able to envision your testing effort as you read through each section.

The issues of Web testing and software testing are very important in today's fast-paced technological society. Many companies, businesses, and private individuals are putting an all-out effort to get a presence on the Internet. It is important that companies and businesses take active steps to test their Web sites since, for many businesses, Web sites will make or break their business. The race to put out a Web site quickly often reduces the quality of many sites. In fact, a lot of frustration and errors can be avoided by hiring a quality test engineer to run your site through a testing process and methodology tapered to the needs of your Web site.

After spending years working in the field of software testing, I have found that there are a limited number of books that cover the scope of Web software testing. This book covers topics that have not been addressed in other books. It is important to me to be able to convey and share with you some tools, ideas, and techniques that I have found helpful.

How This Book Is Organized

This book is organized in a manner that allows for chapter-by-chapter reading and builds a toolkit that will step you through the Web testing process. The book is divided into two parts. Part One, "Managing the Web Testing Process," addresses the methodology and management involved in Web testing.

Chapter 1: The Web Testing Process. This chapter discusses how to test a Web site and how important testing is to the success of your Web site. The presence of online businesses on the World Wide Web has become overwhelming. Because of this, there is a need to identify the testing processes and methodologies that are most applicable to your business. Since testing a Web site is unique and must follow a certain process, this chapter will walk you through the test process.

Chapter 2: Testing Methodology. Testing methodology is an important, but often forgotten, aspect of the Web testing process. A well-designed Web site is essential to the success of a company. It is important to understand the test methodology and carry out this methodology to ensure that all aspects and needs of your Web site are met. It is this well-planned and thorough testing effort that will address the different aspects of the Web design, such as different Web browsers, competing technologies, and variances of the Internet.

Chapter 3: Web Site Management. The management of software projects has always been difficult, but the Internet has added a higher degree of difficulty to these projects. In order for a business to be successful on the Internet, the management for designing and planning the Web site has to be strong. The management must be able to answer critical questions and deploy a plan that is suitable to all involved. Chapter 3 will present ideas, questions, and suggestions to strengthen your management process.

Chapter 4: Risk Management. The quality assurance and testing of a business' Web site are driven by the needs of the business. Business needs drive the issues of risk management and contingency planning. Web site risk management is a process within itself that helps determine how an organization will be affected by exposure to risk on the Internet. Risk management can be used to minimize, control, or eliminate exposure to risks. IT managers

can follow risk management procedures to gauge security concerns as they pertain to an e-commerce site. Chapter 4 will identify some of the risks and present ideas and scenarios that will strengthen your overall understanding of risk management.

Part Two, "Web Testing Tools and Techniques," addresses testing tools and techniques that will aid you in the testing process. It is designed to give you an idea of the test tools that are available, what they do, and how you can contact the companies that offer them. Part Two also addresses the different types of testing that you will need to do, how to do it, and how to document each phase of the test process.

Chapter 5: Web Site Testing Tools. This chapter will introduce you to different Web tools and discuss how to evaluate the tools for your testing effort. It will show you how a particular Web testing tool should be evaluated based on the objectives that you have set up in your Web testing plan. In today's Web environment there are many different types of testing tools and each tool performs different tasks. It is important that you have an idea of what you need to consider when choosing these tools.

Chapter 6: Preparing the Web Environment for Testing. This chapter will explain how a Web site is considered a type of client-server system. Since a Web site is a client-server system it must be tested on the client side, then the server side, and then as a whole. The environment that revolves around the system is important to the overall performance of your site.

Chapter 7: Testing Languages and Databases. Since there are many different components that make up a Web application, it is a challenge to test a Web site. This chapter discusses how environment, network, database, language, and browser interface components need to be accessed and tested. Web technologies

such as HTML, Java, JavaScript, and VBScript, along with databases, input, and output, are some of the Web tester's major concerns. Addressing each technology and component will enhance the understanding of the Web test process.

Chapter 8: Testing on Different Platforms and Servers. Since many problems that current Web sites face have nothing to do with development, but rather deployment, it is important to understand servers and platforms. This chapter will address the challenge of building Web sites with reliability, scalability, stability, and manageability. As Web sites begin to handle business-critical applications, the management and operational issues associated with Web development become crucial. Chapter 8 will also introduce you to several of the servers that are available for the different needs of your site.

Chapter 9: Web Capacity Testing—Load and Stress. Load and stress testing is one of the most critical components of Web testing. The key to a successful Web site is to have the hardware configured correctly so that it will be powerful enough to meet the required demands. Testing is essential to ensure that the demands of a Web site are met. Chapter 9 will show you that by performing Web load testing you will be able to find performance bottlenecks in your design and setup during the early stages of development. By finding these flaws early in the test process, you will save time, money, and keep users happy.

Chapter 10: Running the Web Test. This chapter will discuss what it takes to run the actual Web test. Your understanding of the process will give you the ability to carry out and run the actual test with the tools and methods you have chosen. You will also be able to decide if you want to use automation to carry out all the tests involved.

Chapter 11: Analyzing the Test Process and Documentation. This chapter will illustrate how documentation is an important part of the test process. The test results need to be analyzed for accuracy. The highest level of testing productivity will occur when you find the most failures with the least effort, which is why you should document and prioritize each level and step of the test process. Chapter 11 will also give you several examples and scenarios to use for creating your own documentation.

Part Three, "Templates," provides the sample templates discussed in the book.

Who Should Read This Book

The general audience for this book are managers, Webmasters and Web developers, programmers, and test analysts in charge of developing and testing appli-

cations and Web sites. It is written so that anyone with a Web application to test can use the resources and information covered.

Managers can use this book to guide them through the management phases of testing, implementation, and deployment. It will help by illustrating the different aspects of managing a Web site testing project.

Webmasters and Web developers can also use this book as a toolkit for understanding the Web test process. Since Webmasters and developers understand the coding, language, and tools necessary to set up a site, they can use the book as a guideline to ensure that their design will coincide with the testing methodology and life cycle of the Web site.

A tester can use this book as a toolkit for starting, carrying out, and completing the test process for a Web site. Testers will find test tool evaluations, testing methodologies, and different forms that they will need to use to document the Web test process. Testers will also find other valuable documentation on the different aspects of the test process.

What's on the CD-ROM

The accompanying CD-ROM includes customizable versions of the templates, test scripts, test cases, and scenarios, as well as a resource listing with links to sample tools.

This page intentionally left blank.

PART
One
Managing the Web Testing Process

This page intentionally left blank.

CHAPTER 1
The Web Testing Process

Testing a Web site is a relatively new concept in the information technology (IT) field. Many businesses will test one part of a Web site, failing to see the importance of testing all the major components. Many businesses have not been as successful as others have because of this lack of testing; therefore, the need to test different aspects of the Web site has increased.

The presence of online businesses on the World Wide Web (WWW) has become almost overwhelming. Because of this, you must test your site if you want to succeed, and to do so you need to identify the testing processes and methodologies that are most applicable to your business. Individuals can purchase just about anything on the Internet such as books, medicine, flowers, and paper supplies. To compete in this market, a Web business must be able to handle the volume, secure purchases, and deliver goods to customers. For this to happen, businesses should take Web testing seriously.

Web Testing Challenges

Understanding the Web test process is essential for deciding how to proceed with the selection of a Web test process, automated test tools, and methodologies. Following are several challenges that need to be considered when deciding on the Web process that is most applicable for your business:

The Web is in a state of constant change. The developer and tester need to understand how changes will affect their development and the Web site test process. As technology changes, testers will need to understand how this will affect them and how they will

handle their testing responsibilities.

When setting up the test scenarios, the tester needs to understand how to implement different scenarios that will meet different types of *business requirements.* For example, is a tester testing a site with graphic user interface (GUI) buttons and text boxes or testing HyperText Markup Language (HTML) code? Simulating response time by pressing buttons and inputting different values will verify if correct calculations are valid. (See the section *Business Requirements* for more details.)

The test environment can be a difficult part of the setup for the tester. You need to be aware of all of the different components that make up the environment; the networking piece can be especially difficult to simulate. The following several considerations need to be addressed (Chapter 6, "Preparing the Web Environment for Testing," will further address these):

- Multiple server tiers
- Firewalls
- Databases
- Database servers

In the test environment, it is important to know how the different components will interact with each other. This is illustrated in Figure 1.1.

When setting up the Web testing environment, special consideration should be given to how credit card transactions are handled, carried out, and verified. Because testers are responsible for setting up the test scenarios, they will need to be able to simulate the quantity of transactions that are going to be processed on the Web site.

Security is a constant concern for business on the Internet as well as for developers and testers. There are hackers who enjoy breaking the secu-

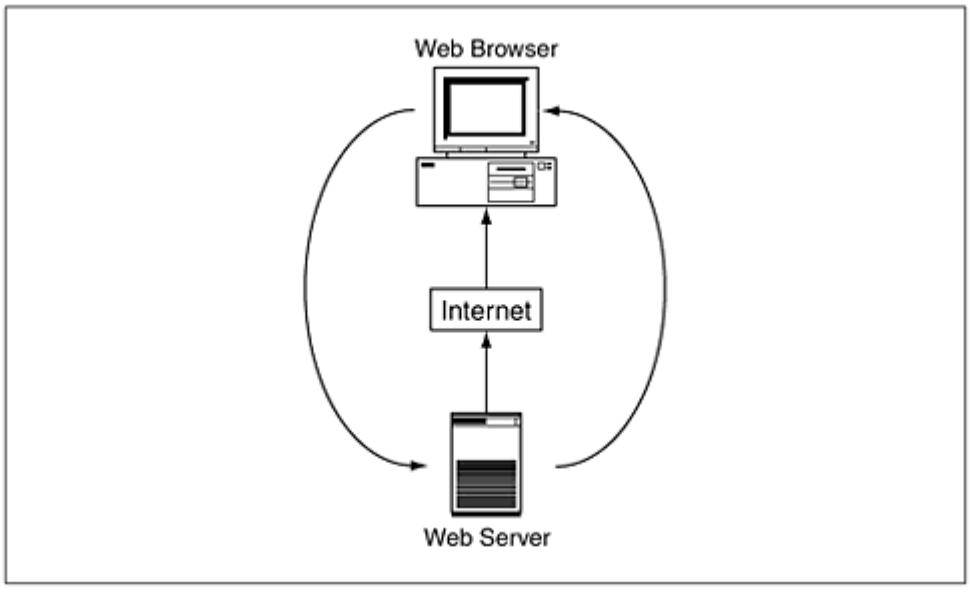

Figure 1.1 Interaction between the web browser, internet, and web server.

rity on a Web site. (We will talk about security as a methodology of testing in Chapter 2, "Testing Methodology.")

This book is a complete Web testing toolkit for testing today's fast paced, highly functional Web sites. It will walk you through the process topic by topic and help you set up the Web test process that best fits your business.

Test Plan Development

The objective of a *test plan* is to provide a roadmap so that the Web site can be evaluated through requirements or design statements. A test plan is a document that describes objectives and the scope of a Web site project. When you prepare a test plan, you should think through the process of the Web site test. The plan should be written so that it can successfully give the reader a complete picture of the Web site project and should be thorough enough to be useful. Following are some of the items that might be included in a test plan. Keep in mind that the items may vary depending on the Web site project.

PROJECT

Title of the project:

Date:

Prepared by:

PURPOSE OF DOCUMENT

Objective of testing: Why are you testing the application? Who, what, when, where, why, and how should be some of the questions you ask in this section of the test plan.

Overview of the application: What is the purpose of the application? What are the specifications of the project?

TEST TEAM

Responsible parties: Who is responsible and in charge of the testing?

List of test team: What are the names and titles of the people on the test team?

RISK ASSUMPTIONS

Anticipated risks: What types of risks are involved that could cause the test to fail?

Similar risks from previous releases: Have there been documented risks from previous tests that may be helpful in setting up the current test?

SCOPE OF TESTING

Possible limitations of testing: Are there any factors that may inhibit the test, such as resources and budget?

Impossible testing: What are the considerations involved that could prevent the tests that are planned?

Anticipated output: What are the anticipated outcomes of the test and have they been documented for comparison?

Anticipated input: What are the anticipated outcomes that need to be compared to the test documentation?

TEST ENVIRONMENT

Hardware:

What are the operating systems that will be used?

What is the compatibility of all the hardware being used?

Software:

What data configurations are needed to run the software?

Have all the considerations of the required interfaces to other systems been used?

Are the software and hardware compatible?

TEST DATA

Database setup requirements: Does test data need to be generated or will a specific data

from production be captured and used for testing?

Setup requirements: Who will be responsible for setting up the environment and maintaining it throughout the testing process?

TEST TOOLS

Automated: Will automated tools be used?

Manual: Will manual testing be done?

DOCUMENTATION

Test cases: Are there test cases already prepared or will they need to be prepared?

Test scripts: Are there test scripts already prepared or will they need to be prepared?

PROBLEM TRACKING

Tools: What type of tools will be selected?

Processes: Who will be involved in the problem tracking process?

REPORTING REQUIREMENTS

Testing deliverables: What are the deliverables for the test?

Retests: How will the retesting reporting be documented?

PERSONNEL RESOURCES

Training: Will training be provided?

Implementation: How will training be implemented?

ADDITIONAL DOCUMENTATION

Appendix: Will samples be included?

Reference materials: Will there be a glossary, acronyms, and/or data dictionary?

Once you have written your test plan, you should address some of the following issues and questions:

Verify plan. Make sure the plan is workable, the dates are realistic, and that the plan is published. How will the test plan be implemented and what are the deliverables provided to verify the test?

Validate changes. Changes should be recorded by a problem tracking system and assigned

to a developer to make revisions, retest, and sign off on changes that have been made.

Acceptance testing. Acceptance testing allows the end users to verify that the system works according to their expectation and the documentation. Certification of the Web site should be recorded and signed off by the end users, testers, and management.

Test reports. Reports should be generated and the data should be checked and validated by the test team and users.

Web Testing Processes

The purpose of the Web testing process is to provide a clear and concise description of what needs to be done. Specifics of the process are discussed in the following subsections.

Objectives

The objective of testing is to ensure that the Web site is ready for operation. The test manager and the testing team have this responsibility. A Web test process will enable a tester or developer to meet critical assignment dates, minimize errors in testing, and improve the overall site.

It is important to realize when you select the process that it forces your test team and all parties involved to follow a precise process of testing. The Web test process builds on a unified process of requirements, analysis of the requirements, development, design, and site code.

If the Web test process is followed accurately, measurable results can be documented and presented to management and eventually the audit team. The *V-process diagram* illustrates the involvement of those associated with the testing process. Figure 1.2 illustrates the V-process; each box represents a different step in the testing process; they are as follows:

Requirements analysis. A specification that identifies the basic requirement functionality of the Web site.

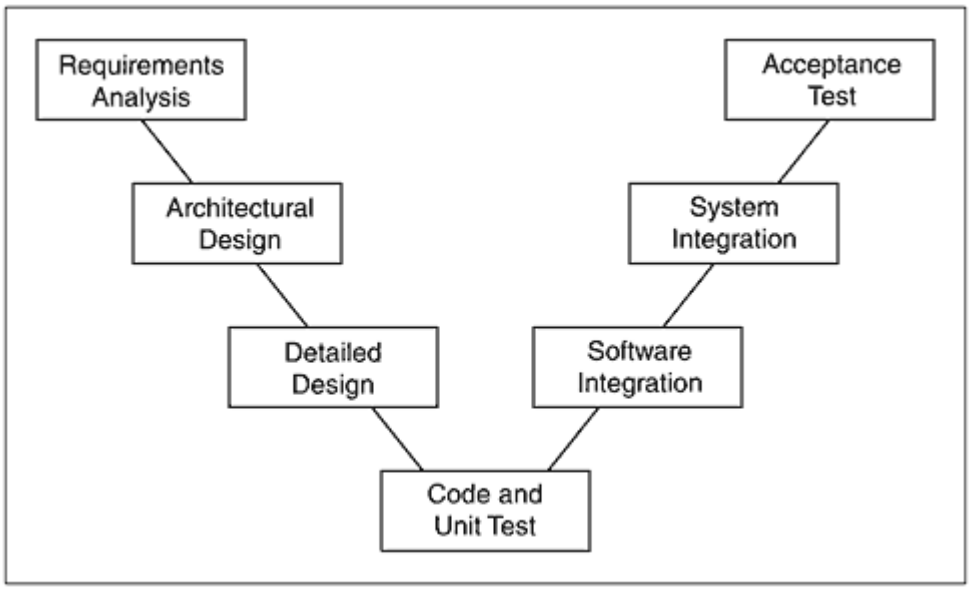

Figure 1.2 The V-process diagram.
www.teleport.com/~qcs/papers/p821.htm

Architectural design. A design specification that directs the designers in developing and laying out the Web site.

Detailed design. A detailed layout of the specifications that shows how each piece of the Web site will fit into place.

Code and unit test. The code is created and a unit test checks that specific segment of code.

Software integration. The process that allows the designer to set up the software and work with the design of the system.

System integration. The process that allows the designer to implement and begin the implementation of the system.

Acceptance test. The final phase of testing allows the user to put the Web site into production.

The V-process diagram is a way to look at the software flow and analyze the development of the Web site. If testing involves the entire Web site, the test cycle begins at the requirements phase and continues through acceptance testing. Testing can occur in any or all of the phases in the V-process diagram; it depends on how thoroughly management wants to test.

The test plan should have quantifiable objectives that relate to testing goals. The level of testing that is performed will give the user optimal Web site performance. This performance depends on server setup, application setup, and the general functionality of the Web site. If the testing objectives and goals are met, the Web site is ready for deployment.

The following are objectives the test manager should consider. To release a quality Web site, each objective should be discussed and met:

System response. Once testing has started, working through the test cycles should be a critical aspect. This is needed to make sure that the system is responding correctly for the particular test. If you are expecting the system to generate five reports a minute and you are only receiving one, an adjustment needs to be made, which may involve data, response time, or even resources.

System availability. When working with users, it is important to make sure that they can log onto the system. When users waste time and energy trying to get onto the system to test the site, it can cause delays in your test schedule and loss of valuable time and money.

Defect tracking. Before testing, a method for tracking problems should be put in place. Whether the developer or tester will log problems will be determined based on the setup of the test team and responsibilities. Problems need to be tracked as soon as they occur and should be tracked throughout the development, testing, and retrofit process.

Deliverables defined. Deliverables should be measured as defined in the test plan. The deliverables are items that can be viewed and measured, such as the test documentation and the verifiable results on the test cases and test scripts.

Web site expectation. The end user should be able to enter data into the system and receive accurate data in return. This is an important concept because the Web site is developed and tested for the user. This is a critical component that affects the objectives; your test cycle should not continue until there are sign offs for each test cycle.

Adequate documentation. A technical writer or a member of the test team should be involved throughout the testing and development process to develop documentation. Each aspect of the Web test process should be supported by documentation such as the test plan, test cases, or even the test scripts. Documentation should also cover the Web site as a whole, such as system, hardware and software, and server specifications.

Goals and strategies. The test team should generate a test plan that will outline critical dates and milestones. The test plan will be developed based on the business requirements outlined by the developers, testers, and end users. Include a list of goals, strategies, and a test approach as a part of the test plan.

Web site platforms. Before testing, each platform (such as mainframe, OS2, or NT) must be configured properly so testing can begin on time. Once the platform is configured, the environmental test team is ready to set up a parallel test to mirror the platform. The platforms are managed by an environmental setup team. Throughout the test cycle process this team will maintain and correct deficiencies in the testing environment as they occur.

Preliminary Testing

Before beginning the actual test of the Web site, you should take care of some preliminary

testing steps to determine overall requirements.

1. Assemble the test team.
2. Prepare documentation. Documentation should include:

1. The business requirements that were accepted by the customer and test team.
2. The full functional design from the developer and programmer (if there are any design problems, the tester will have an idea of the intent and any corrections made).
3. The internal design specifications (if there is a specific internal problem, testers will know how to address it).
4. Any document that may help during the testing (such as previous test results).

1. Create a project plan.

1. The project plan, reporting requirements, and required standards and processes (such as release processes, change processes, and so on) should be available.
2. Identify high-risk aspects.

1. Design a test approach.

1. Set a concrete timetable for each phase of the testing.
2. Set priorities for the order of the tests.
3. Set the scope and limitations of tests ahead of time. Testers need to be aware of what should and should not be tested at this point.

1. Set up an environmental team.

1. Ensure the hardware is in working order.
2. Confirm that the software is ready and the Web site is set for testing.

1. Begin testing.

1. All lines of communications are set up and ready to go. Communication with involved parties is essential; the less time spent looking for people the more time spent on actually testing the software and the Web site.

2. Decide how many cycles your test team will exercise so the environment team can make sure space is available before testing begins.
3. Any tools, such as record/playback tools, should be set up and ready to go.

1. Create test scripts.

1. Test scripts are in place for test tracking. Documentation should be available at the end of each phase so the tester of the next phase is prepared and ready to go.

1. Develop a method to do problem tracking.

1. Create a form for problem and bug tracking so issues can be documented and reported to the appropriate person.

1. Decide how many cycles you want to perform.

1. Establish estimates for each cycle of testing and discuss them with the test team. There should be a determination as to how many cycles will be needed.
2. Create a timeline for each cycle (as well as the entire project). The timeline must be updated and monitored throughout the project.
3. Record each milestone as each cycle is completed.
4. Log and track Web test processes.

As with any new process, follow a systematic Web testing process to ensure all steps are covered. The process should begin by identifying the goals for the site and how the goals will be achieved. A Web site should have an index as its home page and each page that links from that home page needs to be incorporated into the design. Because businesses today are relying more and more on the Web for their success, it is important to understand the development of the site as well as the goals.

Business Requirements

Before beginning the testing project, the tester should have a set of *business requirements* that will help in understanding the functionality of the Web site. Business requirements are a collection of requests and lists from people who have an interest in the project. There are tools available, such as Requisite Pro from Rational, that can assist in the layout of business requirements for testing.

A well-written set of business requirements will outline the goals and objectives for the business and serve as the foundation for your test plan. Business requirements are the

high-level objectives of the organization; these objectives answer the needs of the business as defined by the project's vision and scope.

Business requirements are written after a team of developers, users, and programmers discuss and define the scope of the project. The requirements outline the purpose and implementation of the application and describe the behavior of the system. The team should write the information and develop a representation of the intended purpose of the application. It is from these ideas that the functional requirements are established to accomplish the goals of the proposed Web site.

Following are some ideas for writing effective requirements:

- Keep sentences and paragraphs short and to the point.
- Use the active voice throughout the requirement.
- Check your spelling, grammar, and punctuation.
- Use terminology that is defined in the glossary and data dictionary.
- Do not be vague, and make sure you know exactly what the user means.
- Be detailed enough so the requirement can be carried out.

When a system is designed, users, developers, and programmers work together to decide how the Web site will function. It is from these sessions that requirements are developed and set up based on functional requirements. It is advisable for the developers, programmers, and users to initiate these working sessions as a team to ensure the accuracy of the development.

Business requirements, which follow, should be written with a results-oriented focus:

- Complete (developed as a functional entity)
- Consistent (in line with the design)
- Correct (reflect Web site accurately)
- Nonambiguous (only one meaning)
- Noncompound (stand alone)

The format for business requirements follows:

- Overview and summary of product

- Development, operation, and maintenance of the environment
- External interfaces and data flow
    - Report format
    - User displays
    - Data flow diagrams
    - Logical data stores
    - Data dictionary
- Functional specifications
- Performance requirements
- Condition and handling exceptions
- Priorities
- Modifications and/or enhancements
- Acceptance requirements
- Standards for documentation
- Functional tests
- Performance tests
- Guidelines for design
- Sources of information (for example, white papers, or online documentation)
- Glossary of terms

After the business requirements have been developed, the next step is to decide how to implement and test them using these requirements.

Testing Phases

As the business requirements are established and defined, they will become one of the first phases of your testing process. Understanding their magnitude will help you determine how you will proceed with the Web test. This complete understanding will also help you determine the number of test cycles, test the data used, and set up the test environment.

Communication

Communication is a vital element to make sure that the test team is informed of specific test tasks and changes to the testing process. The test plan is critical in documenting the goals and

strategies for the product and its test approach and will help to determine whether the test goals are handled properly. A good way to track the testing process is to create a checklist to make sure that you are following and completing the test process. Table 1.1 is an example of a testing checklist. Each item in the checklist (or checkpoint) should be a part of the test process and depends on the test life cycle, specification, management, commitment, and communication.

Testing Environment

The test environment should be planned with the number of cycles, type of test data, and the way to test the Web site in mind. This environment consists of the hardware, software, network, and logistics. Early in the stages of testing, it is necessary to determine how this will be set up and deployed. You can set up a test lab or test at the user's workstation. The test data should be selected and kept separate from production data. Remember, backups are essential after each test run.

The hardware should be capable of handling the test load. Users and testers should be given separate passwords for the test environment. The software needs to be correctly configured and monitored for accuracy. The test environment should be set up to handle the correct servers and locations as well as accurate URLs, IP addresses, and any essential server information that is relevant for the test. Testing setup should be readily accessible and the tasks should be easy to perform. If a lab is designed so that it can be used for other projects, cost benefits may be realized.

Resources

Resources are critical components of the testing process. It is important to decide how best to use available resources and how to obtain reliable ones. As a tester, it

Table 1.1 Testing Checklist

QUI	CO	CHECK
		Who can test?
		Why do you want to test?
		Who should test?
		What do you test?
		How do you document results?
		How do you script the test?
		Who develops the test bed?

What is the user's role in testing?

What do you do when you are done testing?

Whom are results reported to?

What happens if the audit team calls you in to answer questions?

When do you see the developers as opposed to the programmers?

As a tester, how do you implement fixes and retest?

How do you track problems?

Who will do the fixes?

How will white papers benefit you when you test the Web site?

Who will write the test plan?

Who is on the test team?

Who sets up the test environment?

Who maintains the product once it is tested?

Will the test environment be used in the future?

Who will need training?

How do we rate the bugs?

Have you done enough research to start the testing?

is important to focus on the goals. Remember that the purpose of testing is to uncover errors in code and Web site design. If you work in an organization, you should focus on several testing items. You need to determine the size of the project and figure out how many resources are required. The test team will be key in determining the amount of risks involved in setting up the testing environment and the testing process. If you have limited testing resources, you will have to determine what parts of the Web site are critical to test before they are released.

Test goals are achieved by working with objectives that are critical to the success of the Web site. Following are questions you should try to answer:

- Does a Web site run as expected?
- Does the Web site meet business requirements?
- Does the Web site run as the programmer designed it?
- Does the Web site run in a timely fashion?
- Is the Web site well documented?
- Are business requirements available?
- Are the design and code available?
- Are definable business requirements set up?
- Are business requirements realistic?
- Has everyone involved agreed on the defined requirements?
- Is the development team aware of the defined requirements?
- Have the users been involved in the development of requirements?
- Is testing performed within a workable schedule so the Web site can be released in a timely manner?
- Is the timetable reasonable?
- Has everyone been informed of his or her role and task with relation to the timetable?
- Does testing involve all platforms that will be applicable for the Web site?

-
- What platforms are involved?
- How many platforms need to be tested?
- Are testers able to uncover errors?
- What are the different types of errors?
- What are expected errors?
- Is there a problem tracking system that will be set up?
- What type of problem tracking system is set up?
- Who will track the problems?

- Does testing involve end users?
- What role does the end user play in the Web test process?
- Who will work with the end user throughout the process?
- Have measurable test specifications been set up?
- How will the standards be measured?
- What will be measured and how will measurements be tracked?
- Will the test include prior measurable results?

Tester Qualifications

A good tester should have a combination of the following skills:

Communication. The ability to convey to the developers, testers, and users the intent of testing and the roles and responsibilities of all parties.

Technical expertise. The ability to understand the Web site and how it works.

Diplomacy. The ability to work well with others and come up with the best solution for the team.

Accuracy. The ability to produce error-free results.

Persistence. The ability to test and retest until an adequate result is achieved.

Of these qualities, persistence is the most important. A tester should have the ability to continuously test and retest without becoming bored or losing focus. Being able to endure this process allows the tester to assure accurate results, and the end result is happy customers.

Deadlines

After the resources and environment have been laid out, it is necessary to set up deadlines to achieve the testing goals. This can be handled by using a project management tracking tool such as Microsoft Project. This tool will allow you to set up a baseline and track your project based on resources, times, constraints, and milestones. The information can be presented in a manner that is useful for management and the test team.

Problem Tracking

Tracking the testing process using a problem tracking tool will produce verifiable test results. There are many different types of tools available (see Chapter 5, "Web Site Testing Tools," for a listing of some of these tools). Some of the features you should include or look for in a tracking tool are date of test error and date of resolution, type of problem and its severity, description and resolution, and problem assigned to/number of days to fix.

Configuration Management

A process that has become important not only to developers but also to testers is a way to track development using configuration management. *Configuration management* is the ability to track and control Web site development and its activities. By integrating problem management with configuration management, you can gain control of development activities and improve quality. For example, if multiple developers are working on the same code at the same time, targeting multiple platforms that support multiple programs, configuration management becomes crucial. In today's development and testing field, problem tracking and configuration management have been combined, which allows for problems and changes to be tracked and assists in the overall process of the Web test.

Web Site Design

The developer/designer of the Web site (who may also be the Webmaster) has the critical function of designing the Web site and maintaining it to the best interest of the business. It is important for this individual to have the expertise to set

up and design a site that will benefit the business. The Web tester must understand the design and have a basic understanding of the language and technology that is used to design the site.

The first important step is to map out the Web site. Mapping out a site involves laying out the design and making sure that each page links to the assigned page. After you map out the site, you should identify all URLs (it's a good idea to set up a database of these URLs). URLs can be assigned a number so that tracking the links during the testing process becomes easier. A test script can be written by going step by step through each URL, and the results can be verified by printing them to an .asp page or a text report (see Chapter 11, "Analyzing the Test Process and Documentation," for an example test script). Mapping out your site will make it easier to develop tests that exercise the Web site design. The components involved in performing the test are HTML code, broken links, spelling, download time, and browsers.

A design layout tool such as Visio can be used to set up a schema of the Web map. Testing tools such as Astra can show the schema of the Web site and its links, giving the home-page address. As the tester gains an understanding of the layout of the pages, he or she will have a better understanding of the way the Web site is organized.

Automated Test Process

A test program that incorporates automated testing will involve a development effort of strategy, goal planning, test requirements definition, analysis, design, development, execution, and evaluation. Because organizations are required to do more with less, automated testing can save time and money.

Automated testing is important to all testing because you can reuse code and scripts and allow testers to standardize the testing process. In the Web environment, automated testing is performed across many platforms, multiple layers of supporting applications, interfaces, databases, and different applications that can serve as a front or back end to the application. Testing in this environment can include the following:

- Functional requirement testing
- Server performance testing
- User interface testing

-
- Unit testing
- Integration testing
- Program module complexity analysis
- Program code coverage
- System load performance testing
- Boundary testing
- Security testing
- Memory leak testing
- Firewall testing

Automation has made these types of testing more efficient and provided more accurate results.

Strategy

The test strategy is an overall approach to testing; it identifies the levels of testing applied and the methods, techniques, and tools that will be used. A test strategy should be adopted by the organization as a whole; it is critical to the success of the Web development and should be detailed in the test plan. Figure 1.3 illustrates the approach to test strategies. Web development, test management, and Web testing are all critical to the overall testing strategy.

A strategy should be developed that encompasses the following:

Software development. A software development strategy should be set up to illustrate and document the flow through the life cycle development.

Testing. Testing will validate efforts, and measurable results, such as number of test cases, number of errors detected, severity of errors, time spent, and which types of testing will be performed, should be documented.

Project management. A workable schedule and timeline should be a major part of the overall test strategy. This timeline needs to be incorporated into

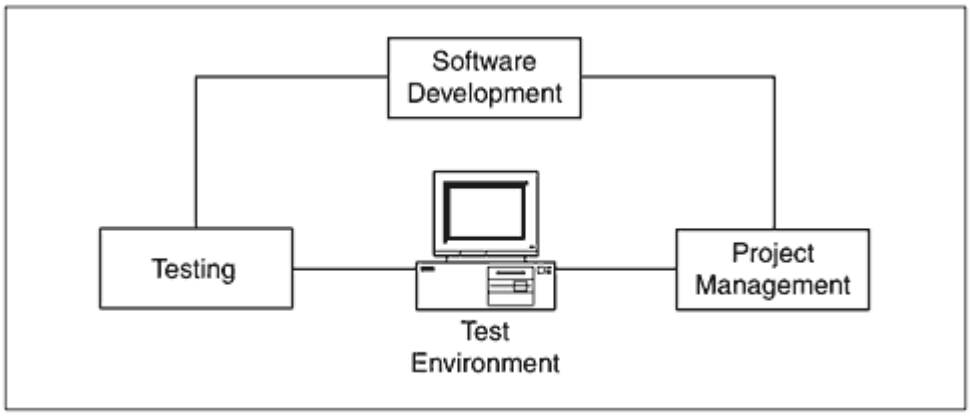

Figure 1.3 Test strategy.

the test plan and published so that all parties involved know their part in the testing process.

Test tools. Appropriate tools need to be selected that are applicable for the application that requires testing. Testing tools should be used if repetitive or a large volume of testing is needed, but it may not be feasible to use automated tools if you are only testing a small application.

Metrics. The ability to track results is important when working through the test process. A decision needs to be made about how measurable results will be tracked and recorded. Several types of metrics that can be tracked are system response time, reliability, efficiency, usability, portability, and number of defects. Figure 1.4 illustrates some of the metrics that will need to be tracked.

Employees. Employees and consultants are a critical part of the test process; how many will be needed should be determined. Employees may also need appropriate training. Testing success is based on the effectiveness of testers and their ability to follow the test process.

Setting Up the Test Strategy

When you set up the test strategy, you should determine the approach of your testing methodology. You should have a strategy for Web site development, testing, project management, test tools, metrics, and employees. You need to

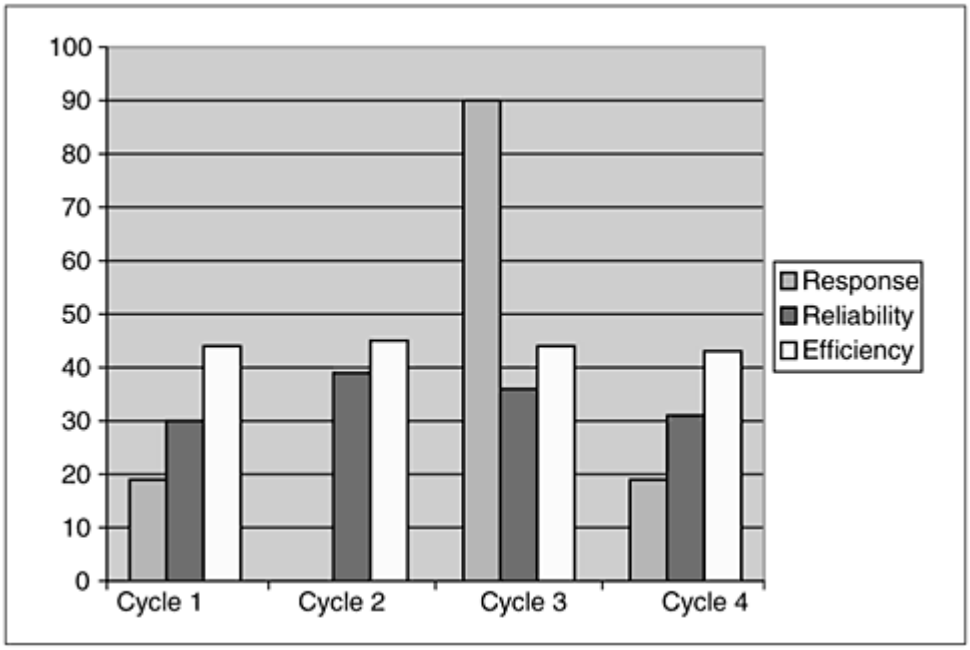

Figure 1.4 Tracking metrics.

decide whether your tests will be manual, automated, or a combination of both. The test strategy will affect the testing team, developers, and end users. The test team is instrumental in the decision process and implementation of the testing process. A test procedure should specify the process and not leave out steps or make assumptions.

Web Test Analysis

Once the Web test process has been put into place, the following information should be analyzed to ensure that the Web test process is accurate and will return required test information:

- The test objectives should be applicable, reasonable, adequate, feasible, and affordable and should follow the needs of the business requirements.

-
- The test program should meet the test goals and objectives.
- The correct test program and test plan should be applied to the project.
- The test methodology, which includes the processes, infrastructure, tools, methods, and planned work products and reviews, should be adequate and updated to ensure that the test program and test plan are done correctly.
- The test work products should be adequate to meet the test goals and objectives.

- Test progress, performance, processes, and process adherence should be assessed to determine the adequacy of the test program and test plan.
- Adequate testing should be performed to ensure test integrity.

Summary

In this chapter we have focused on the Web test process, objectives, strategies, and essential documentation and goals. Chapter 2 will discuss testing methodologies.

This page intentionally left blank.

CHAPTER 2
Testing Methodology

Testing methodology is an important aspect of the Web testing process. Methodologies are derived from a logical system of Web design and problem solving processes that are called *Web development life cycles*. A methodology is an implementation of a logical life process that incorporates the following for each phase:

- Step-by-step activities
- Individual and group roles to be played
- Deliverables and quality standards
- Tools and techniques to be used

A true methodology should encompass the entire Web development life cycle. As with any phase of testing, it is important to remember to write down your test objectives and decide on the type of test data that you will use and how the testing will progress.

There are many different types of testing that can be used to ensure that your Web application is working properly. In this chapter, we'll discuss the following types of testing:

- Unit testing
- System testing
- Functional-black box testing
- Structural-white box testing

- Validation testing
- Verification testing
- Security testing
- Usability testing
- Integration testing
- Regression testing
- User acceptance testing

Unit Testing

Unit testing is generally performed by the programmer, who understands the code. Unit testing should begin as soon as coding begins and should continue through the entire life cycle development. Generally, unit testing does not require formal processes; however it does require a detailed knowledge of the program design and code. Unit testing is usually associated with structural test design because most testers don't have well-defined unit-level requirements to validate.

Unit testing should be handled within a well-designed platform and can be set up using either a specific platform with test driver modules or a test harness. Testing on a specific platform can measure the effects of the unit test on the des-

ignated platform for the Web site. Various tests can be performed by the tester on the system, and these tests can be stored as part of the test process. As it is collected, this information can be stored in a repository for reuse so that tests can be repeated at any time. This is an important component of unit testing.

This type of testing (also called *modular testing*) concentrates on the smallest unit of code. Modular-designed code is tested to ensure that information generated from the code is going in and out of the program properly. It is necessary to test all possible paths into and out of the program.

A *path* is a sequence of instructions or statements that have an entry to and an exit from it. As the test progresses, the path may pass through several junctions, and several processes may pass through each junction. These processes may consist of segments, the smallest component of which is a *link.* This link is a single process that resides between two nodes. The path segment is then a succession of the consecutive links through the same basic path. The length of the path is measured by its links and can also be measured by the number of nodes that are transversed.

The path route may also have distinct loops if any node or link is repeated. A path can have many entry and exit paths. Every path made doubles the number of path values possible. It is necessary to understand the path you want to use and choose that path that will best exercise the link and node. This may involve applying an algorithm that will exercise the path. The key is to select enough paths to achieve the most complete coverage of code. It is

more practical to select many simple paths than to isolate the many possible combinations of complicated paths. The best technique may be to start at the beginning and select the most obvious path through the code. These paths can be built upon as the path test continues.

Because a programmer tests each module (or unit) of code, it is easier to fix errors as they are uncovered. The focus is on the workability of each code module instead of an entire program. For example, if you are writing a program in Visual Basic and each module is tested, you will find errors before the next step is coded and the application is put into production. Taking the time to unit test can save time and money.

A drawback to unit testing is that it requires emulation of the complete program. Unit testing tests specific modules that may need a connection from another tested module. The unit test case should be designed with a functional tie-in to other modules for complete functional unit testing. Each module test is unique to the testing process. This type of testing is crucial to the Web testing process because the smallest test progresses to the next level of testing.

Examples of a unit that could be tested in a particular language are:

- Class module (C + + or Java)
- Function or subroutine (C)
- Individual procedures, functions (ADA)
- Menu or display (4GL)

Keep in mind that a unit test may involve other modules, so the unit test would need to be modified to simulate the missing modules. The test case should address the starting point, the input to the unit, what the test case actually tests, and expected outcomes. To do a unit test the developer or tester must be familiar with the code, whether it is C, C + +, or Java. The unit test should include test cases that are functional and can be reused and modified for other scenarios. The cases and scripts then become a part of your historical test suite. (See Figure 2.1.)

Let's take a look at a snippet of Java code. This is a simple example to demonstrate an important point. The purpose of this unit test is to test for an existing variable that ceases to exist when the program ends.

```
Class ExistTest {
    Int test = 5;
        Void printTest() {
         Int test = 10;
          System.out.println(∀test= ∀ + test);
}
public static void main(String arguments[]) {
   ExistTest st = new ExistTest();
```

```
        St.print test();
    }
    }
```

The test case would be set up with a starting point of the class, the input to the unit, what test case actually tests, and the expected outcomes, as shown here:

Starting point – class ExistTest
Input to unit – int test=s;
Actual test t – input can be printed out.
Expected outcomes – s, 10

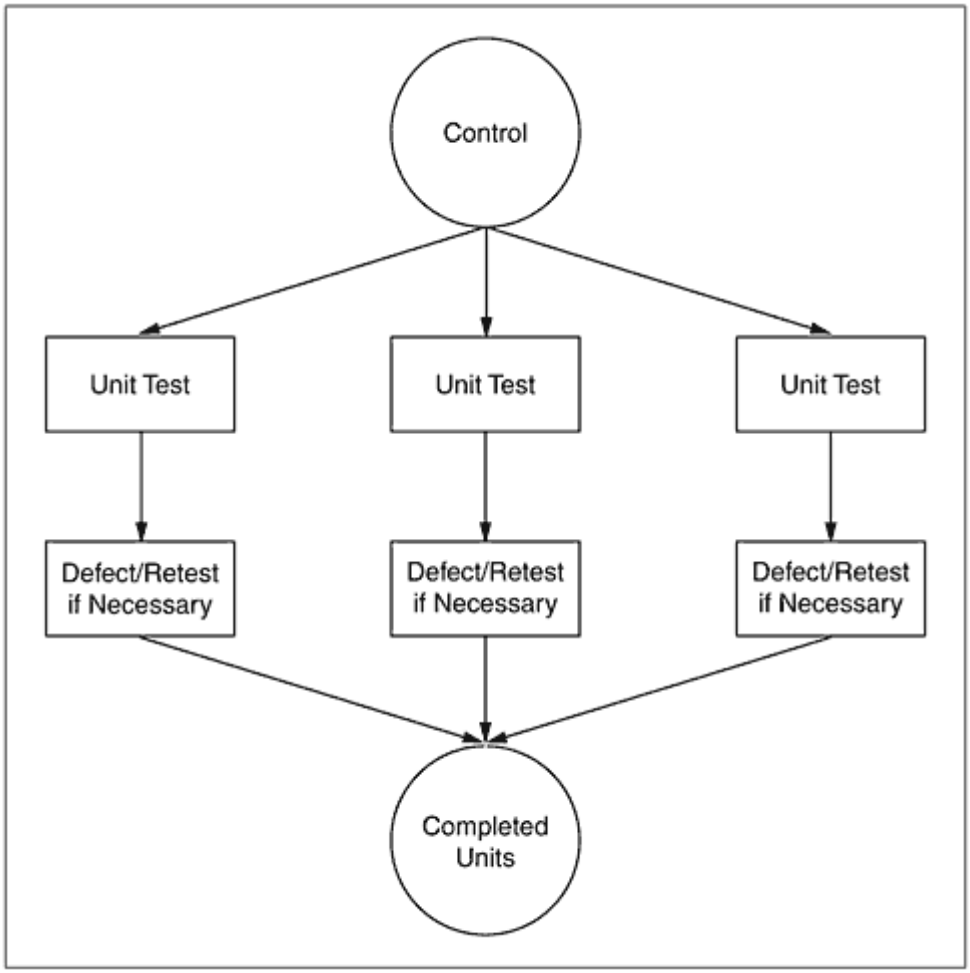

Figure 2.1 Example unit test.

In this example you would use a test to refer to the instance variable and test the local variable. The *instance variable* is designated as int test = s. This is a relative variable assigned by the programmer. The *local test variable* is the test t variable and is used to capture and print the input.

Unit tests are important because they give developers confidence in their code and help

them understand how their part of the unit fits into the entire program.

For an example, let's look at the code to build a Date class in C + + that has the following properties:

- A date can be initialized with a string (YYYY-MM-DD), three integers (Y, M, D), or nothing (today's date).
- A date object can yield its year, month, and day or a string of the form YYYY-MM-DD.
- All relational comparisons are available, including computing the duration between two dates (in years, months, and days) and adding or subtracting a duration.
- Dates need to span an arbitrary number of centuries (e.g., 1600 to 2200).

Your class could store three integers representing the year, month, and day. (Just be sure the year is 16 bits or more to satisfy the last item in the preceding list.) The interface for your Date class might look like this:

```
// date.h
#include &string>
#include ∀duration.h∀ // a three-int struct
class Date
{
public:
   Date();
   Date(int year, int month, int day);
   Date(const std::string&);
   int getYear() const;
   int getMonth() const;
   int getDay() const;
   string toString() const;
   friend operator&(const Date&, const Date&);
   friend operator&=(const Date&, const Date&);
   friend operator>(const Date&, const Date&);
   friend operator>=(const Date&, const Date&);
   friend operator==(const Date&, const Date&);
   friend operator!=(const Date&, const Date&);
   friend Duration duration(const Date&, const Date&);
   void addDuration(const Duration&);
};
```


You can now write tests for the functions. You may want to use and implement a test

such as the following:

```
int main()
{
   Date mybday(1951, 10, 1);
   test(mybday.getYear() == 1951);
   test(mybday.getMonth() == 10);
   test(mybday.getDay() == 1);
   cout && ∀Passed: ∀ && nPass && ∀, Failed: ∀
       && nFail && endl;
}
/* Output:
Passed: 3, Failed: 0
*/
```

The function test maintains the global variables nPass and nFail. The only visual inspection you need to do is to read the final score (the actual output of what passed and what failed). If a test fails, an appropriate message would print.

NOTE When you progress with the testing, you can build test suites that can be used to keep related classes for future testing. For more information, see "Advanced Solutions for C/C + + Programmers," [Chuck Allison, *C/C++ Users Journal*, September 2000, 48–61 (www.cuj.com/code)].

System Testing

The system test verifies that the design and the system satisfy the requirements. An effective system test requires a concrete and testable system-level specification. A complete system test plan can be designed and prepared from documented *use cases.* A use case is the full functionality of the system. The importance of each use case is the frequency of its operational profile (a modeling operation that takes into consideration the functionality of the system). The operational function is taken directly from the business requirements. Because the behaviors that are used most frequently would have a greater potential for system failure, they are considered the most critical to the system test.

Following are important components of the system test:

Goals of the Web project. Once goals have been established, it is necessary to test to make sure they have been achieved.

Major functions. The users should determine what the major functions of the Web site are for the application. Examples of major functions for a Web site are logins, feedback forms, anticipated load, and any calculations the application may use.

General inputs/outputs. When a user signs into a Web site, specific information is requested before the user can enter the site. Once users are signed in, they expect certain information to appear. Examples include correct logins, accurate account information, and

updated user information.

Performance. When the application is activated, the user expects a site that is easy to use and will provide the expected information.

Growth. A site should be built to grow to accommodate anticipated software and hardware updates.

Operation and environment. The user should be able to activate the site through any browser. The site should be able to accommodate ASPs, equivalent frames, text, PDFs, or other required specifications.

Compatibility, interfaces. If a site is designed as an intranet application, it is possible that it can be independent of the user's PC platforms; however if it is an Internet application, the platform compatibility will be dependent on the server.

Table 2.1 shows how an example transaction system would be tested.

Black Box (Functional) Testing

Black box testing (also called *functional testing*) is a widely used software testing technique. It indicates that the component under test has inputs, outputs, and a specification, which states the relationship between inputs and outputs. Black box testing implies that the selection of test data and the interpretation of test results are performed on the basis of the functional properties of a piece of software. A third party should perform black box testing because the programmer

Table 2.1 System for a High School Transaction System

COMPONE	SYSTEM REQUIREMENT	TEST
Goals	To replace existing handwritten documents.	Test to ensure all handwritten documents are obsolete.
Major functions	Login, social security number (SSN) check, sign up for classes.	Test a login using a smart SSN, sign up for classes, check to see if classes are updated in real time.
General inputs/outpu	Input SSN as login; output will show existing classes.	Test to see if SSN will activate system; once the system is activated, the user can see output and make changes to input as deemed necessary.
Performance	How long does the system take to process a transaction?	Test to see how long it takes to activate system, check peak times, and check data integrity.

Growth	As the amount of work required by the system is increased, can the program grow with the increase?	Test the system to see if it can operate with an increase in load.
Operation and environment	A student can use a personal computer at home or on campus and will be able to access the program through the WWW.	Test the system through a test computer from within the system and then from outside the system.
Compatibilit interfaces	The application will work on all the systems and will work as part of the intranet on campus and the Internet from outside the campus.	The system will be tested though the intranet and then through the Internet.

has inside knowledge of how the program works and is least likely to discover any errors. The tester should have knowledge of the user requirements; black box tests do not have to involve the participation of users.

Black box testing is considered the basis for most testing practices. It is used to determine if the program is functioning as specified. There are different approaches to black box testing. One method is to test each program feature or function in a logical step sequence. Another approach is to test module by module (this is similar to unit testing).

NOTE Keep in mind that black box testing assumes that the tester does not know anything about the application that is going to be tested. The tester needs to understand what the program should do, and this is achieved through the business requirements and meeting and talking with users. When applying black box testing, it is important to understand the business requirements so that the test cases can be set up.

After a general understanding of the application is achieved, the tester needs to understand how information is input into the system and the expected output. This type of documentation is gathered through *test cases.* A test case is a document that lists the inputs to the system and expected results from the system.

Following are several functional features you should consider:

Browser functionality. Is the browser compatible with the application design? There are many different types of browsers available and the greatest challenge a tester faces is making sure that all browsers have been tested. The most popular Web browsers are Netscape's Navigator, Microsoft's Internet Explorer (IE), and America OnLine (AOL).

GUI design components. Are the scroll bars, buttons, and frames compatible with the browser and functional? It is important to check the functionality of the scroll bars on the interface of the Web page to make sure that the user can scroll through items and make the correct selection from a list of items. The buttons on the interface need to be

functional and the correct hyperlink should go to the correct page. If frames are used on the interface, they should be checked for the correct size and whether all of the components fit within the viewing screen of the monitor.

There are many types of black box test scenarios. Let's say we are testing a function that has a single integer parameter and the function should return the value of 1 if the parameter is greater than 100 and 0 otherwise. The function test could be set up like the following:

```
int More(int);
int More(int a);
 {
     if (a > 100)
       return 1;
    else
    return 0;
}
```


In the unit test we tested a small component of code to make sure it performed as designated by the developer. In the functional test, a specific condition is tested. In our example, this condition will continue to loop until one of the conditions is met (in this case the input can be either greater or less than 100). A test case should be developed and designed with a particular condition in mind, such as a possible function that performs an expected output. When setting up the test case, incorrect inputs need to be included as well as correct inputs to examine all possible routes and functions for the program.

White Box (Structural) Testing

White box testing (also known as *structural* or *glass box testing*) is different from black box testing in that it tests the full knowledge of the application's inner workings. White box testing goes hand in hand with the unit test and is generally used by developers as well as testers. The objective is to test the components by exercising every path through the code.

When testing a Web application, for example, you would test the results of logging in by using different user IDs. The statement that would handle this could have two possible paths: one to see if the value was true, and another to see if the value was false. As you progress, you can limit the number of paths and reuse the cases and scripts.

Testers need to test the objects and methods from the white box perspective, ensuring that every avenue through the code exhibits the correct behavior. A general method would be to create a small test scenario and test the specific object or method to make sure the structural design is working efficiently. Generally a test harness or a stub is created and kept for testing reuse. A *stub* simulates called routines not yet developed to allow routines to be tested. It delivers a message and returns a reply. A *test harness* should include the following:

- A standard method for setting up the test and a cleanup after the test
- A method for selecting individual tests or all tests

- Ability to analyze the output and expected results
- A form that will include any errors reported during the test
- A specific language used for the test harness written in any language

A driver can simulate calling a routine that will allow a called routine to be executed. It is also considered a program that will test each module. The *driver* is the opposite of a stub; it simulates the called component and the environment from which the component is to be tested. This allows for modification without recreating the entire scenario.

The basic difference between white box and black box testing is that black box testing can mask problems that white box testing would uncover. For example, black box testing might indicate correct functionality because the object problem is not visible. White box testing would reveal incorrect objects and methods through the specific testing path set up to follow the object or method through the path. White box test design allows the tester to look inside the black box and focus on the internal information that makes up the software and to steer the selection of test data.

In white box testing, the program structure is used to set up the test cases. By using the program structure you can set up a test program flow graph, which can help set up the design of the test cases. By using a test flow graph, you can exercise the code by way of a path from entry to exit and exercise the code at least once. Testing control structures of a procedural design in a white box environment should consist of the following steps when applying them to test cases:

1. Within a test case, set up independent paths that are exercised at least once.
2. Make sure logical decisions are exercised for both true and false paths.
3. Make sure loops are executed at their boundaries and within operational bounds.
4. Make sure internal data structures are exercised to ensure validity.

Using the preceding steps, you will exercise all possible paths of the test and ensure that logic errors are addressed and the paths are correct. If errors are found, you can make design corrections.

Validation Testing

Validation testing is a way to make sure that the system is supporting and reacting correctly to information that is entered into the system. It incorporates a series of testing techniques to guarantee correct functionality of the system. This

was one of the key types of tests used for Y2K compliance and validation. Web testing validation is absolutely essential to the success of a Web site. These tests are designed to

make sure user expectations are fulfilled. The overall effort for validation testing is to begin testing the smallest possible unit of the Web application (the units and modules) and to work through integration testing (discussed in the section *Integration Testing*), which will make sure all units and modules are working together.

Validation testing should be used to begin the early processes of the software life cycle test. The aim of validation testing is to demonstrate that the process fits into the life cycle and to measure results so there is an acceptable output. The software life cycle is an important process for the tester and developer to understand when setting up the validation test. Following the life cycle process, you assure the customer that the application meets or exceeds the customer's expectations.

There are four phases to the software development life cycle:

- Planning
- Analysis and design
- Implementation and testing
- Installation and maintenance

Validation testing is used to uncover errors and verify the functionality of the test. Typical validation activities incorporate different types of methodologies (some of which are covered in this chapter), such as:

Unit testing. Validating the smallest unit of code for correct output.

Usability testing. Validating that the application is usable for the owner of the application.

Function testing. Validating that the application works within the specifications of the business requirements.

System testing. Validating correct system requirements that are used to produce valid results.

Acceptance testing. Validating tests for users for accurate responses to the business requirements.

If you do not want to break down all the validation tests yourself, you can use a validation test suite tool. Validation test suite tools will provide quick test results for all Application Program Interface (API) and Application Binary Interface (ABI) specifications. The purposes of the test suite tools are to validate a product's functionality with the customer requirements. Validation tests should have definable results, that is, what you expect to see from running the tests and the ability to rerun the tests. If the user or tester would like to see the test run again, the actual and expected results should match.

Verification Testing

Verification testing reviews an application. There are many different types of reviews that can be used to verify an application. As with validation testing, this is best done in the software development life cycle process. All those interested and using the application should participate. This is done through routine meetings, inspection of the process, and time monitoring. Verification begins by going through the business requirements and ensuring that the requirements are functional and usable for the application. The developers should verify the functional design so that the testers can use this as a basis of setting up the test cases and scripts for testing. The developers are key in this verification process to ensure that the functional and internal specifications are working accurately before the testers use the correct information.

Security Testing

Security testing is considered one of the most important methodologies. Customers need to feel confident about ordering on a Web site so that they will return to the site and purchase other items. A test scenario should be designed to exercise the site's security controls. This test will make sure that the installed system security is responding correctly. A scenario should be designed that will deny services to customers, deny password access, deny access to certain areas of the site, and browse secured data. A tester's two main concerns are *network* and *payment transaction security.*

Network security is best tested through an automated test tool. A test tool has the ability to test deeper by challenging user rights, passwords, logons, logoffs, and password expirations. Testing should ensure that when a password is given, it is kept secure by encryption. Three parts to the security test-

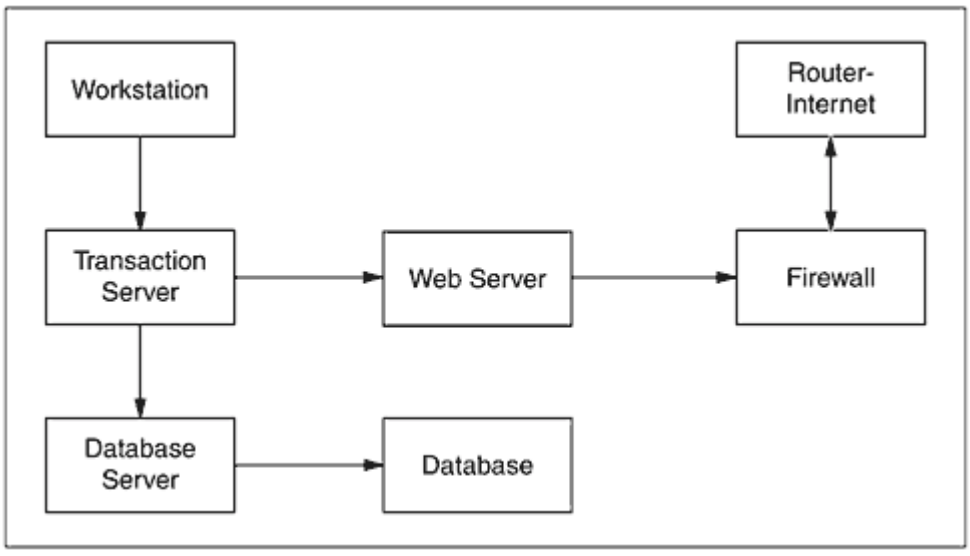

Figure 2.2 Web server security.

ing process are at the browser level, server level, and the connection component. There are two types of server security: *server-side* and *client-side.* Client-side security protects the user, and server-side security protects the Web server (see Figure 2.2).

Usability Testing

Usability testing is necessary to reduce the number of people that leave your Web site prematurely. Usability testing is important to ensure that users can find the information that they are searching for within a click or two. Performing usability testing will keep your users happy and increase return rates to your site. Usability testing tests the user's ability to navigate and find information and should begin early in the development cycle when changes in design are easier to make. Usability testing should also involve comparison testing between your site and other similar sites.

An important component of usability testing is making sure that the site looks appealing to the viewer and information is easily accessible. Following are several questions you need to ask when preparing your usability test:

- Are menus easy to navigate and located in an easily accessible area?
- Is there a home button or link on every page?
- Is the user required to scroll to see the whole page?
- Are all targets, where a user must click, big enough for ease of use?
- Are there too many pop up windows?
- Is data persistent if a form is filled out and then the user goes back a page, or will the user have to refill all of the fields when returning to the page?
- If there is a file to download, is the size of the file displayed to the user?
- If a plug-in or component such as a viewer or audio player is required, is the user given a link and instructions to download it? For example if you have links to PDF files, is a link provided to where the user can download Adobe Acrobat?
- Is the contact information for your organization, including an email link, available?

Remember to test for accessibility issues, such as providing alternative text for people who use devices that convert text to speech or providing a text-only site in addition to one that includes graphics and frames. You can send your URL to www.cast.org/bobby to make sure that your site is accessible to those with disabilities. Bobby is a free service provided by the Center for Applied Special Technology (CAST) to help Web page authors identify and repair significant barriers to access by individuals with disabilities.

Integration Testing

Integration testing combines components of an application and tests into a complete working application. With unit testing you test individual modules and units; with integration testing you will now take those individual units and modules and test how they work together. The following are the four different types of integration testing:

Structure testing. This type of testing is very much like white box testing discussed earlier in the chapter. This type of integration testing will exercise the path of the inputs and outputs and check for the valid modules and objects.

Functional testing. This type of testing is very much like black box testing discussed earlier in the chapter. This method of testing exercises the functionality of the requirements.

Performance testing. Tests the actual integration of the application and how the application will perform under testing.

Stress testing. Tests the system's limits. This is also considered load testing.

The primary difference between integration testing and system testing is that system testing focuses on behaviors and bugs that are a part of the entire system. Integration testing concentrates on the interaction between modules and objects that work together. By the time you get to this level, you are validating that all units and objects are working together correctly. As you place all of the objects or modules together, you may see other errors that need to be corrected. You may want to use incremental testing and test from the top down and bottom up through the application and code.

Incremental testing is used during the unit test and is a step-by-step process that tests each module segment. The programmer starts at the top of the application and tests each unit downward. Each statement in this test must be passed before going to the next step.

Regression Testing

Regression testing is applied to customized software to provide assurance that modifications act as intended and do not adversely affect the behavior of unchanged code. A retest approach to regression testing attempts to reduce the time required to retest a customized program by reusing tests and retesting the customized program. Regression testing attempts to test for changes made to an existing system, and it is important to be able to reuse the test cases and scripts that were prepared for earlier testing. It has been discovered that when regression testing occurs, additional errors are uncovered; thus, regression testing is time

consuming and expensive. It is possible that new code changes can contain new bugs and cause the application to fail; therefore, this type of testing is essential.

User Acceptance Testing

User acceptance testing takes an entire application and reviews the overall workings of the application. As the tester moves through the process, user acceptance testing will fulfill the requirements, assist in the training for the application, and also help in the development of documentation and guides for the program.

Testing at the user level can be approached differently from testing at the system function

level. User acceptance testing attempts to bring together all the components of the system, and the user will provide their input and either accept or reject the system. User acceptance testing and system testing are similar in that coding and grammatical errors can be checked at both levels of testing. The system is fully tested by the users against the requirements, which are defined in the analysis and design stages, and the corrections are made as required. The users and the developers should work together to include their specific expertise in the functionality that is being tested; this will ensure that the system will work properly.

The users along with testers and technical writers will document the results of each test and generate error reports describing the problems discovered during testing and forward them to the developers. A formalized error tracking system should be designed to allow a clear line of communication to occur between the testers and developers so that errors can be corrected. After the user and testers have verified that the system is working correctly to their specifications, a formal sign-off is required. This form can be used as formal documentation and kept for auditing.

Remember that user acceptance testing is the last stage of testing performed on a system. During user acceptance testing, users will find faults with the system that will require regression testing. This is the last stage for individuals to test before the application is released into production. It is important for the users and testers to have enough experience to adequately test and conclude that the application is working correctly. Users are an important component in user acceptance testing because they understand what they want the application to do and they were most likely involved in the business requirement process. Remember: Users have a genuine interest in using the application. Although

users may not have the technical expertise in testing, they understand the application and know how it is to be used.

Information on additional methodologies not covered in this chapter can be found at:

www.sqa-test.com/webtest.html

http://directory.google.com/Top/Computers/Programming/Software_Testing/

www.dmoz.org/Computers/Programming/Software_Testing/

www.mtsu.edu/~storm/literature.html

www.sqe.com/stareast/c_sessions_1.html

www.badsoftware.com/outsourc.htm

www.sdtcorp.com/trntest.htm

www.datatech.com/products/seminar/5102out.htm

www.microsoft.com/technet/ecommerce/testpr.asp

Summary

An important concept to remember when you are choosing a Web site testing methodology is

to select methods that are applicable and relevant to your specific situation. Web site testing is not an easy task. It is filled with constant changes (as is the Web). As soon as you get a Web site in place, technology changes. After launching a Web site, you may find that it is quickly outdated and needs to be upgraded frequently. The testing methodology and practices need to be designed to keep up with changes in technology and to be easy to maintain.

Chapter 3 will discuss Web site management.

This page intentionally left blank.

CHAPTER 3
Web Site Management

Software projects have always been difficult to manage successfully, but the Internet adds a higher degree of difficulty to these projects. For a business to be successful on the Internet, the managers who design and plan Web sites have to be knowledgeable about updated Web testing and Web software and hardware. One of the reasons Web sites do not succeed is that management does not take testing seriously and finding problems is left to trial and error.

Becoming an Internet Business

Most Web sites start as an idea—a business wants to increase its existing business sales by presenting an Internet presence. Once the idea has been put into action, many issues need to be addressed before the plan, design, and strategy can be implemented. A few questions need to be answered before beginning to design and plan a Web site:

- How will the Internet site benefit the business?
- Does the company or individual have the resources needed to carry out the project?
- Has a qualified Web project manager been chosen?
- Is a test team in place and does the team understand the importance of automated testing and ecommerce?
- Does the business have an adequate server and network?
- Has a person been assigned to create an ebusiness design-quality model?
- Will the business be able to handle an increased influx of business that Internet presence may bring it?

Strategy

Managers should answer all the preceding questions and then decide how to best create a Web site. The project management team will plan, design, test, and implement the Web site according to the needs of the business. Before the team can be assembled, the business leaders should meet with the Web project manager to discuss the following strategic concerns:

- Where is the business today?
- Where would we like to be tomorrow and what is the ideology behind the strategy?
- What tactical plans need to be created to help adhere to the proposed strategy?
- What are we missing, what gaps do we need to fill to be successful?
- Who is the competition, can we compete with the competition, and what can we do to be better than our competition?
- What is the paid-out cost versus the initial payback in terms of anticipated payback?
- What are the one-, three-, and five-year plans?

-
- What services do we offer as a business that could possibly evolve into other industries and businesses?
- Can our business handle the anticipated increase in business and will our site be able to handle the load?

Once the business leaders and project manager have agreed on the path that the business will take, a team needs to be selected to carry out the desired strategy. One of the key team members and probably the most overlooked person of the Web site management team is the test lead (or tester). The tester is unique to the Web site management team because a true Internet presence has many components that need to be tested before the Web site can be put into production. The testing can be done with automated test tools, and the tester should be trained and have an expertise in the chosen tools. We will talk about different automated test tools in Part Two of this book, "Web Testing Tools and Techniques."

Design Quality

The next step is to design the Web site according to design-quality factors. The quality of the design needs to be assessed up-front, at mid-cycle, and at the conclusion of the project. Through each phase the metrics for the project need to be established and monitored. The metrics will measure certain parts of the design and management of the project. Table 3.1 describes the quality factors of the Web design and the essence of the factors.

Management Team

Project management focuses on management processes that can be used to organize and work through a project assignment. The following people may be part of the project management team:

Project manager. The leader of the team, who guides the process, resolves issues, and is in charge of the decision making.

Client, user, or customer. End user who will actually use the product, a vendor that is preparing the product for its customers, or even a supervisor.

Table 3.1 Web Design-Qu Factors

DESIC	ESSENCE OF FACTOR
FACT	
Correc	The extent to which program fulfills the objective and goals of the end user.
Efficie	Optimal amount of computing is used to perform the specifications that the program was designed to satisfy.
Flexibi	Effort required for modifying an operational program or a hardware environment.
Integri	The extent to which program satisfies its specification and fulfills the user's objectives.
Interop	The ability of heterogeneous computer systems to communicate and cooperate in problem solving.
Mainta	The effort needed to locate and repair an error in an operational program or hardware component.
Portab	The ability of a programmatic unit or hardware component to operate on multiple hardware and software platforms, without having to be reworked.
Reliab	The extent to which a programmatic unit or hardware component is to perform according to its function with precision.
Reusab	The extent to which a programmatic unit or hardware component can be reused in other software or hardware design solutions.

Testab The effort required for testing a programmatic unit or hardware component to ensure that it performs according to its specification or anticipated load level.

Technical specialists. A technical specialist may be responsible for network security. They should have an expertise different from everyone else on the team and will not directly be a part of the testing process. We will go into more detail as to how they will be chosen and what their role is later in the chapter.

Financial advisor. Inclusion of a financial advisor will depend on the size of the company and the project's budget.

As we stated earlier, personnel may change according to the scope of your project. A large project would require a lot of people, whereas a small company may only need a person or two to maintain and carry out the project.

Web project management is a collaborative process. Success depends on working and communicating effectively with individuals who have an interest in the plan. It does not matter if the individuals are members of the project team or executives interested in high-level project status information. It is the project manager's job to follow the project through and release the updated version to the customer. A Web project manager must have several different types of skills, such as the following, if the project is going to be successful (see Figure 3.1):

Communication. The manager must be able to talk to management, the test team, and the customer and be able to articulate to everyone involved what needs to be done and how the process will take place. Successful Web project managers are those who can recognize the importance of the people with whom they work.

Web project management. The project must be managed with three issues in mind:
Cost
Schedule
Quality

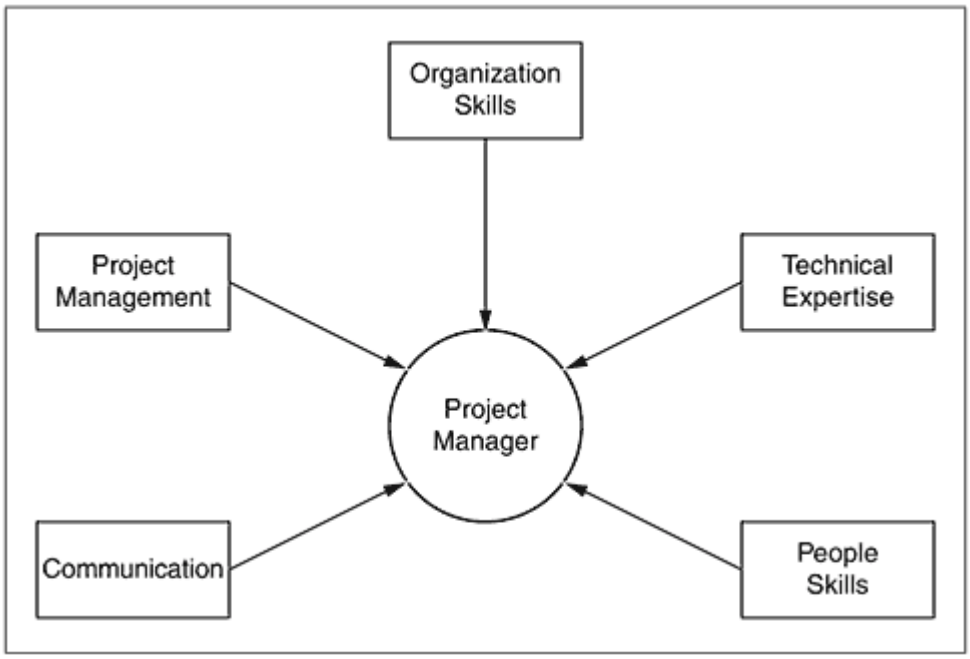

Figure 3.1 A project manager must possess certain skills.

These three items must be blended in a way that will give the customer the best possible product when the test process is complete.

Ability to organize. It is important to set up a formal organization chart. A good Web project manager will use a good management tool that has the flexibility to help organize, change, reorganize, and complete a project.

Technical expertise. One of the reasons a Web project manager is chosen is technical expertise. The manager may have been a programmer, test engineer, or developer before venturing into the world of project management. Managers know how the software life cycle works and what it takes to complete a project from beginning to end. They use this experience and knowledge to guide the team through the software testing phase and to work with the test team to decide what needs to be tested and how it should be tested.

People skills. The Web project manager recognizes that people play a vital role in completing the project on time, within cost, and with top-notch workmanship.

The role of the Web project manager is to oversee and provide instructional input to the project. The manager must make sure that the project goes according to plan. He or she creates a product by balancing three factors:

Quality. The product must be delivered to the customer according to agreed upon standards.

Time. The product must be delivered within a specified time. If adjustments need to be made, they need to be justified and agreed upon.

Cost. The total cost of the product must be agreed upon. A financial limit should be set up

and the product delivered should be within these limits.

The main objective for the Web project manager is to set up the project and actively participate in the project as it undergoes changes. He or she needs to make sure that the plan is realistic and the project plan milestones are attainable. It is important to keep the team organized and focused on the goals that have been established with the project plan. The key is to manage the team and time so that it stays focused and on task and achieves the final goal. The next section discusses how the Web project manager keeps everything together.

Meetings

It is essential during the development and implementation of the project that you, as the project manager, schedule several types of meetings to keep everyone informed of responsibilities and tasks. It is important to create an agenda and timetable for each meeting. Following are several critical meetings that should take place:

Introductory meeting. The introductory meeting will establish your role in the project. At this meeting you should introduce the members of your team, including area of expertise, why team members were chosen to work on the project, and how, as a team, they will set up and follow a project plan to develop and test a software package. You should also go through the schema so that everyone understands how the Web site will be tested. This meeting will set the tone for all subsequent meetings.

Regular status meetings. The purpose of status meeting is to see the progress of the project. These meetings should be scheduled at a time that is agreeable to everyone who needs to attend. As the Web project manager you are going to have to juggle schedules and times. Before the meeting you should send a memo to all attendees that specifies what will be discussed and what each member should bring to the meeting. You should have an agenda for each meeting that includes a list of the major topics in order of importance, the status of each phase, the target schedule, the actual schedule, and bottlenecks that have caused problems. The meeting should also include announcements, feedback, and additional discussions and potential problems. The meeting should start and stop on time, and it is vital that you stick to your meeting schedule. During the meeting each member of the team should be asked for input and should contribute to the overall meeting. A good meeting is a productive meeting at which relevant issues are dealt with and tasks are evaluated and ranked.

Review meetings. Review meetings evaluate the workmanship of the Web site. At each meeting you will access one component of the product and evaluate whether the output of the Web site meets the requirements that were set in the business requirements. Usually this meeting only involves the customer, the Web project manager, and the team members who are working on that phase of the Web site. This meeting should always be recorded so that the information, guidelines, and changes are all documented. The minutes from the meeting should then be distributed to all team members to keep them abreast of ongoing concerns and acceptance

of the Web site. The review meeting should be scheduled at each critical point and a demonstration should be prepared for the meeting.

Management meetings. The management meetings are set up to evaluate the schedule and the cost and quality of the Web site. These meetings should establish how the project is progressing and if any drastic changes need to take place.

Project planning, the next phase to be discussed, creates the actual project timeline and a plan that allows you to track the project through its various phases.

The Project Planning Phase

The first step in the project planning phase is to understand what the project is, what resources you will use, and what it is that you want to accomplish. You should take an overview of the Web site from different perspectives and talk to the developers to understand how the Web site will be designed and how it will work. After talking to the developer, you will have an idea of the size and functionality levels of the testing. Once you know what you must accomplish, it is imperative to have a firm, written objective or mission to share with the team. This will enable you and the team members to have something to evaluate your progress against and also will help to keep the team focused.

To test a Web site, a Web project manager needs to put together a test team and define the team's roles and responsibilities for the project. The test team should meet with the project manager and put together a realistic project plan. The Web project manager should then take ownership of the project plan.

Figure 3.2 shows the different components for which the test team is responsible.

The Test Team

Now it is time to select a test team, which will take some time. You must identify the needs of the project and determine what each individual will be responsible for testing. You also have to decide who will be on the test team and what types of expertise will make the project a success. The team should be taught how to use the tools that are needed to build and maintain the Web site. The roles and responsibilities of the test team will be identified in the project plan. It is the job of the test team to carry out each facet of the project plan that

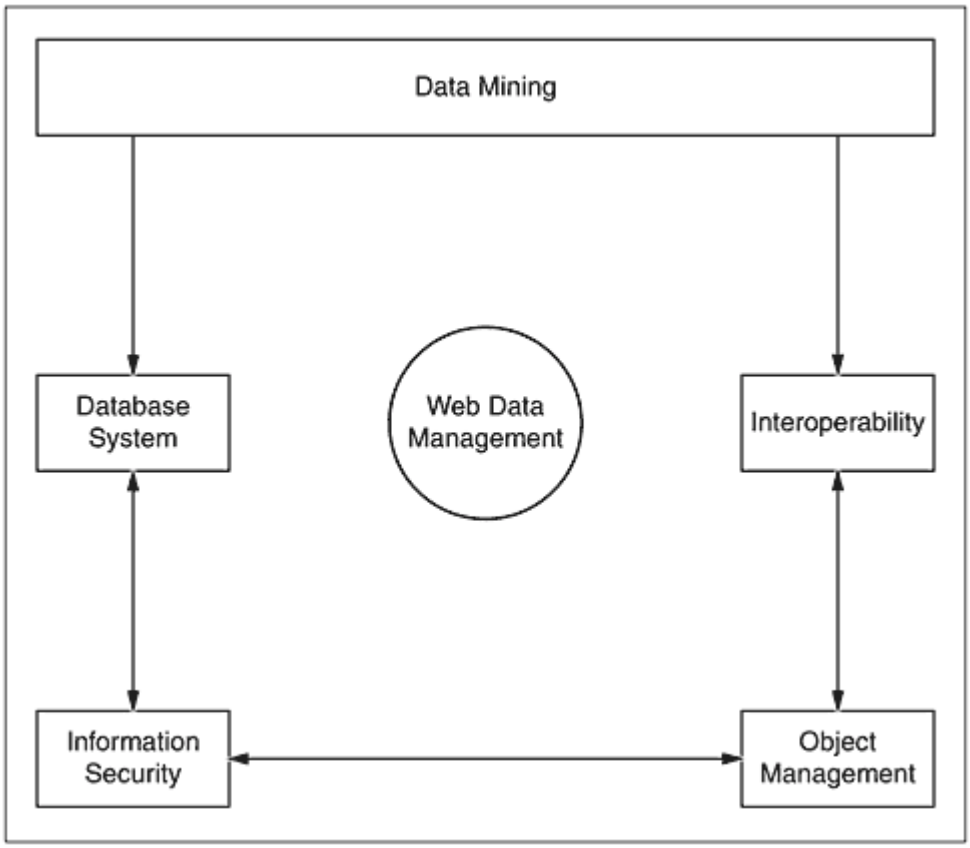

Figure 3.2 The different components of Web data management.

pertains to it. As each task is completed, the responsible member of the team should report to the Web project manager or test lead. To test the Web site you will first need to look for a quality tester. A software test engineer is a trained, skilled individual. There are a few recognized institutes that certify that testers have the necessary testing skills; a certified tester who has expertise in the area that you are testing is a good choice. If your project is based on a client-server, you should hire a tester with experience in that area. If it used a mainframe, you should have a mainframe tester. In other words the level of expertise should be a determining factor when hiring testers. The tester will have the opportunity to work with the following people (see Figure 3.3):

Developer. The developer has the ability to design the Web site and to communicate with the test team. The developer must also understand the

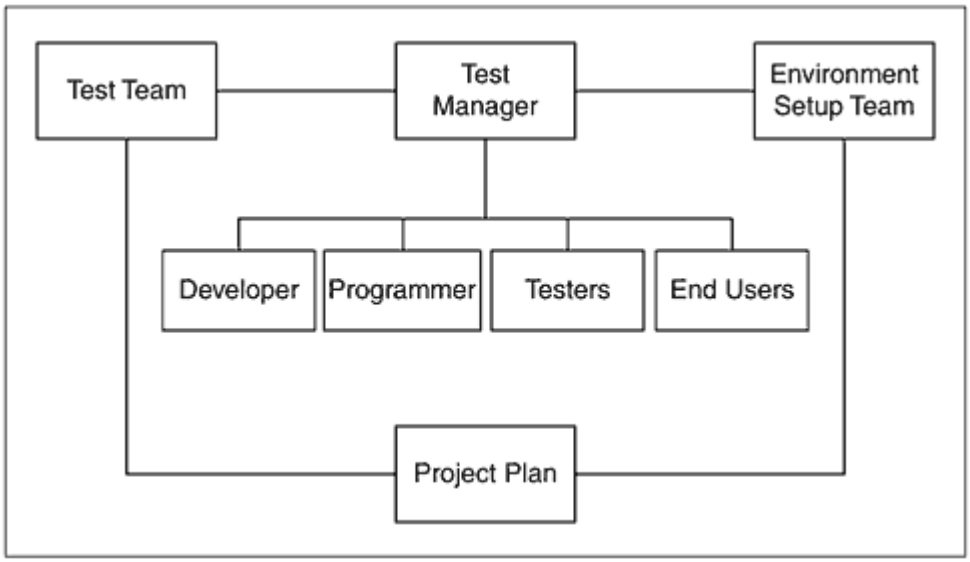

Figure 3.3 Possible test team members.

complexities of the Internet and how unit testing will be done according to the design of the Web tester.

Programmer. The programmer has the ability to code and understand functionality of the application and the Internet.

Web tester. The Web tester needs to understand functionality between programmer and end user, the importance of testing the Web process, and the use of automated Web test tools.

Technical writer. In addition to understanding the complexities of an Internet application, the technical writer needs to take technical information and work it into understandable documentation.

End user. End users must make sure the business requirements fulfill their needs.

The customer and contractor can be useful team members, depending on how they are used:

Customer. As the Web project manager, you will have to work with people outside your project management team. The first contact you will make is

with the customer. The customer can be your boss, the end user, or an outside vendor that has hired you to develop and test the Web site or application. It is essential that you keep the customer informed of the project as you are tracking it. The customer should be aware of all anticipated problems and date changes.

Contractor. The other outside contact you may need to make is with a contractor (or consultant). You may want to hire the contractor to do tasks that members of your team are not qualified to do or to enhance your workforce. The contractor should be made to

feel like an essential part of the Web test team.

As a project manager, you will need to assign people to meet with the end users to go through the Web site step by step. Because we are focusing on Web site development, the customer is our user. The end user knows the application better than any other member of the team; it is important that his or her needs are addressed and all problems are taken into account.

In a large project a single individual could not manage all the issues involved. The project leader or manager will have the following responsibilities and must make sure that each one is handled appropriately:

Organization. Throughout the project many of the dynamics and resources will change. The Web project manager must keep track of everything and make sure that all other team members are keeping their areas organized and tracked. During status meetings the Web project manager must bring all this information together.

Customer support. The Web project manager must assure the customer that the project is being carried out the way it was designed and discussed. The business requirements must be met and the Web site package must pass all the testing. The Web project manager must work with team members to make sure that the customer is kept informed of the progress and all the customers' questions are directed to and answered by the correct team member.

Technical support. Throughout the project you will need technical support. You must make sure that the developers and programmers understand the nature of the project. They will need to use their technical expertise to assure that you will deliver a quality product. You should also know what

type of outside technical support you can receive from others who have built previous versions of this Web site project.

Project management support. For some managers this is a difficult aspect of being a project manager. You are in charge of this project and you will need support at times, but some project managers feel that if they ask for support or assistance, it will reflect poorly on them. However, the more support you get from your management and the more you keep it informed of all issues, the better it will be for all of you. Each member of a management team has a different expertise, and being able to draw on this from each other will be beneficial to everyone involved.

Business Requirements

The project manager needs to be able to analyze the business requirements (see Figure 3.4). For example, you need to make sure that the product you are developing will provide the proper information to the end user. On the financial side, you need to make sure that the application will generate the essential financial reports and make sure the end users know how the reports will be generated. The reports should pertain to the specific type of business for which they are designed; if the reports need to be customized, the end user needs to understand how to customize them. Differences between new and old reports need to be

discussed with the parties involved. Make sure that the data end users receive will still be applicable with the new Web site.

Business requirements also come into play when you start talking about the hardware and software that the Web site will use. It would not make sense to develop a Web site if those who will use the new site do not have the appropriate software or hardware to run the new program. The present hardware should be able to support the new software; if it is not capable of supporting the software, has the expense of upgrading or replacing the hardware been considered? If your present system will support the software, does it have enough memory to support growth? As the hardware is upgraded and replaced in the future, will it have any bearing on the software? Once you have resolved all the hardware issues, you need to be sure that you have the right software interface on your machines to support the new software. For example, if you have a Windows-based environment on your machines, will the new software be able to interface in a Windows environment?

Page 59

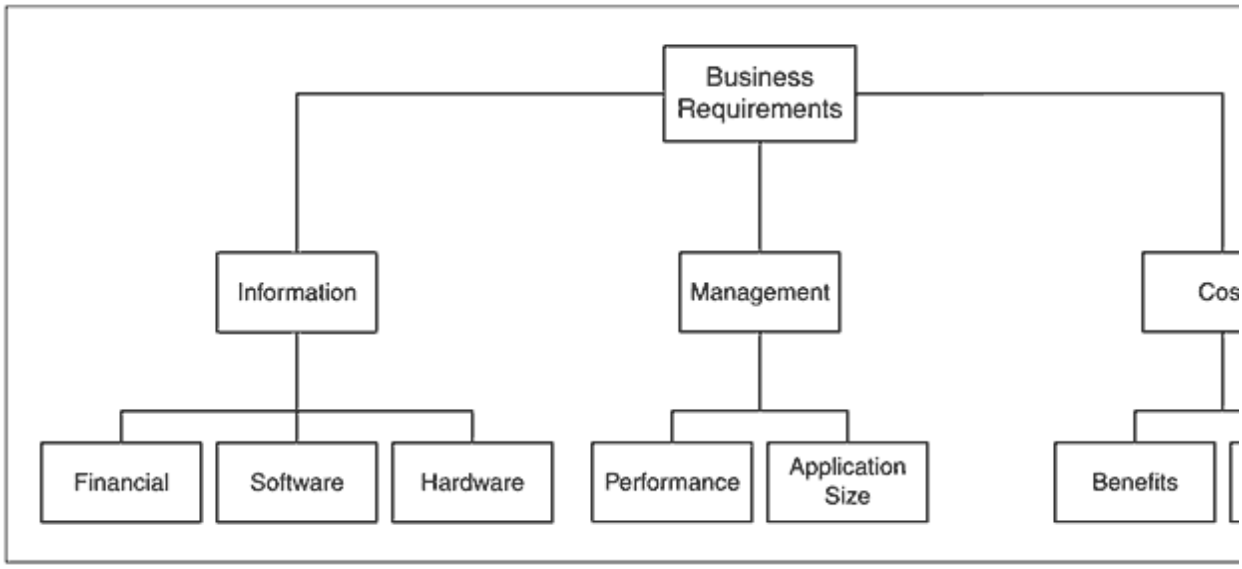

Figure 3.4 Business requirement items.

Page 60

Creating the Project Plan

You will now need to take all of the information that you have gathered and put together the actual project plan, which will identify the resources, the test team, and the risks. If you have done a good job researching the preliminary information, you should be well on your way. The following issues need to be identified and discussed:

Planning phase. Who will be involved in the planning phase?

Knowledge of the Web site. What is the purpose of the software?

Audience. Whom is the Web site targeted to? How many people will be able to access the Web site at once?

Timetable. What is the time frame for the project? What happens if you fall behind schedule?

Testing. What needs to be tested? Who decides what needs to be tested?

Aspects of the Web site that need testing. Will all new features need to be tested? Which of the older features did we enhance and how do we want them tested?

Project tracking. What project management tool will best suit your needs? Who will track what phases of the project?

Once you have answered these questions, you are ready to write the project plan.

Preparing the Project Plan

At this point, the Web project manager will put together a rough project plan to discuss with the project team. It should be designed so members of the team know their roles and responsibilities for the overall project. The Web project manager oversees the project plan and adjusts the tasks according to the strengths and weaknesses of the overall team.

The project plan is considered the mission of the project. To achieve success, everyone needs to share the overall responsibilities of the project. The purpose of

the mission is to test the Web site and also to minimize the risk of the Web site's failure. It is the responsibility of the test team to look for problems and isolate, track, and resolve them. The Web project manager will assign each of these tasks to the appropriate team member and then will make sure the team works as a unit. This will assure that a quality product will be delivered to the customer.

The project plan should also include:

Flexible scheduling. The manager needs to precisely plan and track the project to manage all the details of the team. There are many different types of management tools available; a good one, such as Microsoft Project, includes a scheduling feature that provides useful information. Project management software should be easy to learn and use.

Clear communication. Project management is a shared process. It is essential to work and communicate effectively with individuals who have an interest in the project plan, whether those individuals are members of the project team or high-level executives interested in the project status. Project managers can assign tasks to members, inform them of tasks, and inform them when differences in the original plan occur and changes in scheduled tasks are made. Some of the test management tools can also interact with other applications; for example, Microsoft Project allows you to interact with the Microsoft Office applications.

Deciding on a Time Frame and Project Tasks

After you have identified the resources and testing issues, it is time to set up a time frame for the testing. When you are setting up the time frame, you need to keep in mind the scheduled

start date, completion dates, and the availability of your key players. It is critical to have realistic time frames. Time frames can best be managed by using a project management tool, which lists the critical phases of your testing milestones and assigns workable target dates. It also tracks your project's progress, prints reports, and allows management to identify the potential for bottlenecks or problems that could arise as you move through the project.

Once you have defined the main testing tasks, you can then break the tasks down into subtasks. A good management tool will allow you to set up the tasks

and subtasks. A management tool allows you to lay out major tasks, resources, time frames, and milestones and then allows you to break tasks down even further into subtasks. As you begin your project, you will be able to fine-tune the project plan. Setting up a baseline will help you evaluate the progress of the test project.

Tracking Progress

Good Web project managers use a tool that allows them to measure the success and growth of the project. There are many different types of charts, graphs, and milestones that will illustrate what managers need at their fingertips. These graphs and charts should be a part of every status meeting so everyone on the test team will know the status and progress of the project.

You can create pie charts, bar graphs, 3D charts, and other types of graphics with most test tools. These illustrations can provide the test team with an effective model for the project's progress and a measurement of which phases of the project need more attention. Once again, the more materials you have to evaluate and measure your progress, the easier it is for you to motivate your test team. The following are examples of different graphs and methods:

Critical path method. Critical path method (CPM) examines the order of, and time required for, the various tasks that are required to complete the project. Each task has a start and a finish date.

Pert chart. A Pert chart is a flow chart that illustrates the relationships between the tasks necessary to complete a project and their order of importance (see Figure 3.5).

Gantt charts. A Gantt chart lists the tasks and projected completion times on the left-hand side. On the right-hand side the duration of each task is represented graphically, showing the start and end dates and any dependencies and resources that exist. The Gantt chart is a clear representation of all the phases of a project, and the milestones can be graphically displayed (see Figure 3.6). At any given time the current status of the project can be found by looking along the horizontal axis for the day in question. You can then assess the project and the tasks that should be in progress. The tasks to the left have been completed, and those to the right have not yet started.

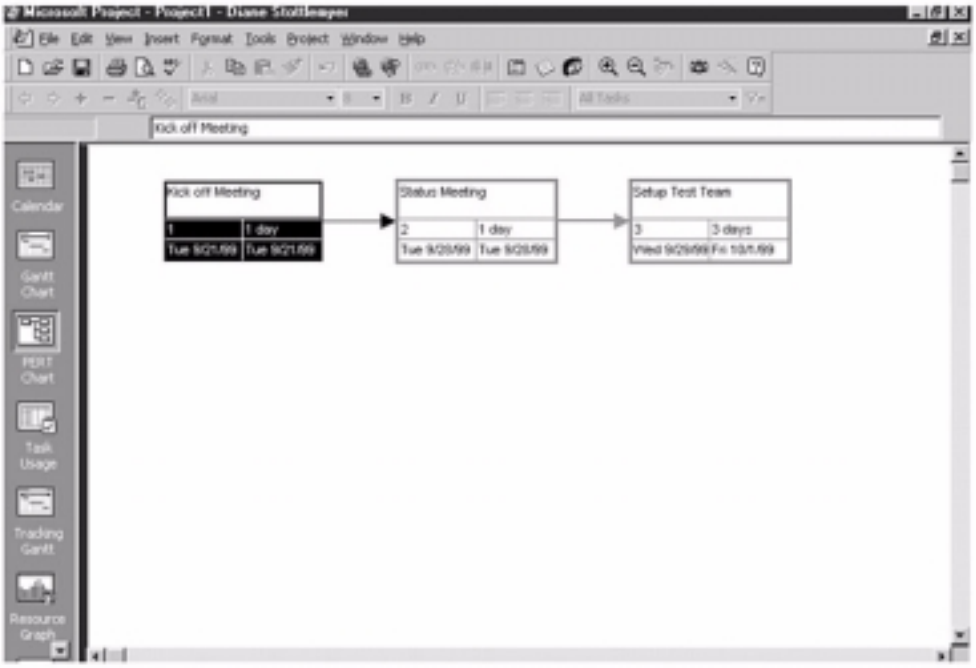

Figure 3.5 An example of a Pert chart.

Identify the Resources

The resources involved in Web site testing include the hardware and software; however, individuals are the key resources that contribute to the overall success of the testing project. One of the most critical steps in assuring the success of the Web site testing effort is to identify, hire, and recruit quality testers. At this stage, it is beneficial to meet with all relevant managers and key personnel to help identify what the critical components are and what personnel are involved. Generally during this phase you will identify any needed software and hardware, other special considerations you will need to test, and the actual members of your test team. You will also identify the different levels of testing.

Depending on the size of the application, you may need an environmental team that sets up a testing environment. You may need just one machine with

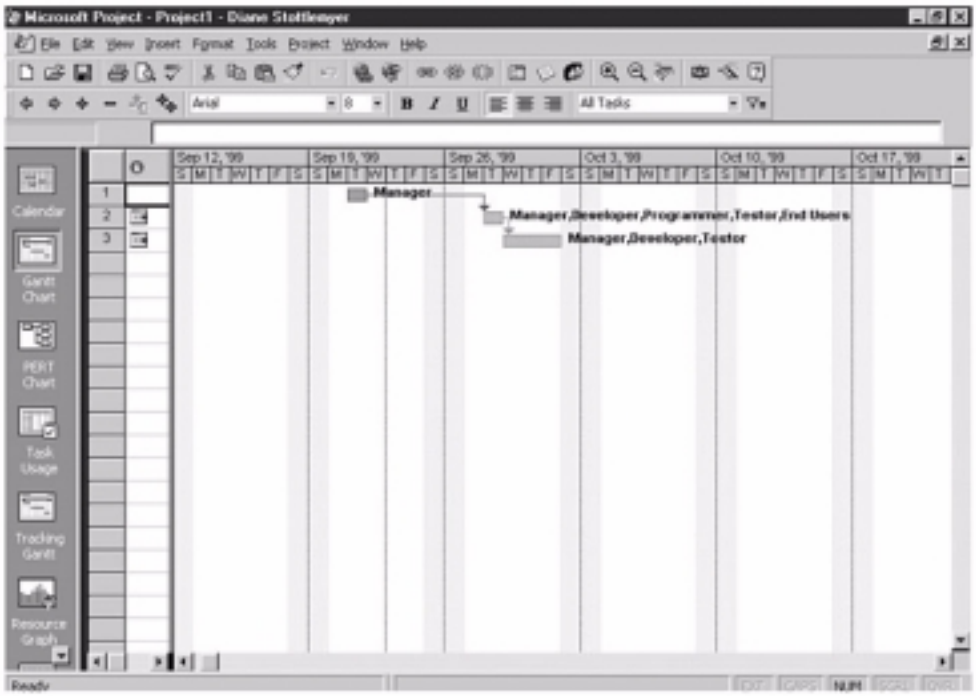

Figure 3.6 An example of a Gantt chart.

one tester or an entire test lab. The testing environment depends on several components, including, but not limited to, the following:

- Platform
- Operating system
- Version release

An environmental test team has a different concern than the test team does. Its specialty is setting up the environment so the test team can come in and run the tests in an ideal environment.

NOTE At this stage, allocating resources for the project will affect individuals working on your team who have other ongoing projects. Planning the selection of the team to coincide with your timetable is critical to the success of the project.

Concluding the Project Planning Phase

It is important to review the project plan with everyone on the project team and to achieve a general consensus. Discrepancies should be discussed at this point and appropriate changes made. It may even be a good idea to circulate the plan to everyone so that the people involved have a chance to evaluate and revise it. After reviewing your project plan, you need to think about the first steps in start-up testing.

From your testing experience you should ask yourself a few questions:

- Why do you lay out a test project plan?
- How do you predict which types of testing will follow in sequence?
- Where will you conduct the test?
- Who will conduct the tests?
- How will you report the test?
- What are you testing?

You must remind your test team throughout the process that they all have a role in improving an important piece of software. Team members should be motivated to test the product as if they owned it. The more time you take to test the Web site, the better chance you have of releasing a high-quality product on time.

Web Site Management Tools

We now know how the Web project is going to be set up and who is going to set it up and are ready to talk about the actual tools that will be used to manage the Web site as it is being developed. There are several tools on the market and it is important to pick the tool that will work best for you. If you use a tool like Microsoft Project 2000 as a general tool, you should make sure the tool you chose for the Web site will track the same type of information.

Table 3.2 includes products that contain site version control tools; combined utilities and tools; server management and optimization tools; and authoring, publishing, and deployment tools; and significant site management or testing capabilities.

Table 3.2 Tool for Site Managem

TOOL	DESCRIPTION	REQUIREMI	WEB SITE ADDRESS
BladeRunn	Web content design, creation management, and publish tool from BroadVision Inc.; enables companies to create, manage, and publish econtent for its Web-enabled applications.	Uses XML as its technology backbone and MS Word for content creation.	www. interleaf.co

JetStream	Site management suite for Web server monitoring, link checker, real-time client usage analysis; logs file analysis, problem determination, performance measurement, and load balancing.	Can manage any Web server on any platform if server is either located on same NT system or same network segment with JetStream. For WinNT, it is installed as a service and accessed via any standard Java- and JavaScript-enal browser.	www. winddancer com
WebReady Manager	Web position form analyzer and promotion tool from Monocle Solutions; checks sites' search engine rankings.	Requires Win95/98/NT.	www. monocle-so webready/ webready. shtml
Web Site Garage	Services for maintaining and improving Web site. Automated site maintenance checks, optimize graphics, and analyze traffic. Free single-page on-the-Web tune-up or fee-based services for entire site. Services include load time check, link check, link popularity check, spell check, HTML design check, browser compatibility check for 18 different browsers, platforms, and screen sizes.	Web site.	http:// websitegara netscape.co
EPrise Participant Server	Web site management tool from EPrise, for content and workflow	Runs on NT or Solaris.	www. eprise.com

Table 3.2 (*C*

TOC	DESCRIPTION	REC	WEB SITE ADDRE
EPri Parti Serv (con	management, access and version control, and business rule management within a single platform.		
Web SiteI	Web site management tool from ParaSoft; includes publishing manager, link checker, HTML checker with custom rule selection, orphan file checking, and more.	Web site.	http:// thewebki com/ products/ webking/ quick/ about.htn
Web Site Dire	Web content workflow management system with browser-based interface; includes configurable workflow management, email submission of Web content, and email notifications; allows defining and applying existing workflow and approval rules to Web content management process.	For NT, Unix	www. cybertear
Equa	Load balancing server appliance and site management tool from Coyote Point Systems. Web-based interface for load balancing administration, server failure detection, real-time server monitoring of server response time, number of pending requests, etc.	Web site.	www. coyotepo

continues

Table 3.2 Tool for Site Managem

(*Continue*

TOOL	DESCRIPTION	REQUIREMENT	WEB SITE ADDRESS
WebTrend Enterprise Suite	Web site management tool including log analysis, link analysis and quality control, content management and site visualization, alerting, monitoring and recovery, proxy server traffic analysis and reporting.	Web site.	www. webtrends.c products/ suite/ default.htm
e-Monitor	Web tool from RSW for 7 × 24 Web site monitoring. Can use test scripts created with its e-Tester tool; allows wide range of corrective action and notification responses. Includes a wizard script generator that generates scripts in standard Visual Basic. Evaluation version available.	For Win95/98/NT.	www. rswsoftware com
HotMetalP	SoftQuad's Web development tool for Web site authoring and development and management; includes capabilities link management, site mapping.	For Win95/98/NT.	www. softquad.co index_main html
Unicenter TNG with Web Manageme Option	Site management application from Computer Associates; includes access and security control, monitoring, logging, metrics, server management, and network management.	For Microsoft and Netscape Web servers.	www.cai.cc
Interwove Team Site	Web development, version control, access control, and publishing control tool.	Works with many servers, OSs, and platforms.	www. interwoven. com

Table 3.2 (*Cont*

TOC	DESCRIPTION	REQUI	WEB SITE ADDRESS
Intra Solu	VitalSign Software's suite of products for dynamically monitoring faults, response times, congestion, downtime, bottlenecks, timeouts, performance changes for intranet systems.	For NT.	www. vitalsigns.co
Blue	Web site analysis and management tool; includes link checker, site mapper, reporting, statistics, integrated FTP, and uploading.	For Win95/I	www.exit0. com/ez1/ products/ pro2000.htm
Pow	From Electrum Multimedia, for customizable automated site mapping, HTML validation, link checking.	Win95/I and MSIE 3.0 or later.	www.electru co.uk/mappe
Site	Freshwater Software's product for site monitoring and maintenance. Runs on servers and monitors server performance, links, connections, logs, etc., and provides notifications of problems. Includes published API for creating custom monitors. Monitors mimic users' end-to-end actions.	For NT or Unix.	www. freshtech. com
SITI	Web site management and editing tool collection from Greyscale Systems; includes link checking, global search and replace. Checks for orphan files, calls HTML editor from within program.	For Win 3.1/95/N	www. morning. asn.au/ siteman

continues

Table 3.2 T for Site Manag

(*Conti*

TOC	DESCRIPTION	REQ	WEB SITE ADDRI
Astr SiteI	Mercury's Web site management tool; scans Web sites and highlights functional areas with color-coded links and URLs to provide a visual map of site. Site map includes HTML, CGI scripts, applets, etc. Shows broken links, access problems, compares maps as site changes, identifies usage patterns, and validates dynamically generated pages. Change management/tracking.	Evalu copy availa For Win9	www.he mercury interacti com/ product
COA Web	Coast Software, Inc. WebMaster site management tool; for Web site file management, link checking, site version comparisons, page download timing, and estimating, server log file reporting. Includes HTML editor and file manager, page display verification, global search and replace.	For Win9	www.co com
Blac Trac	BlackBoard Software's Web project management tool; shareware available from theDownload.com site in the Internet Site Management section. For tracking Web project's code, etc., including program name, info, creation date, related files, comments, documentation, version, status, programmer, changes, and reporting capabilities.	For Win3 or better	http:// downloa com

Table 3.2 (*Co*

TOC	DESCRIPTION	REQUIREM	WEB SITE ADDRES

SiteI	Opposite Software's tool for Web site management; includes meta tag management, spell checking, diagnosis, repair, optimization, global search and replace, HTML-to-text converter, link checker, HTML validatior; integrated HTML editor and browser. Updateable rulebase.	For Win95/NT.	www. siteboss.cc
Oper	Interwoven's configurable control system for deploying from development to production environments; includes automated deployment, security, and encryption capabilities.	Rollback capabilities if used in conjunction with the company's TeamSite product.	www. interwover com
Tear	Interwoven's collaborative Web site production control, administration, and management product for enterprisewide Internet and intranet projects; includes version control, browser interface, comparison capability, file edit merging, and variable lock controls.	Client side requires NS 3.01 + or MSIE3.01 +; server side compatible with many available Web servers.	www. interwover com
Dyn	Inso's Web site publishing and development management product for multiuser control, version control, link verification, and site deployment; includes remote access, multiple editions, and deployment. Capabilities include remote access, multiple editions, dynamic publishing.	Web site.	www.ebt.c

continues

Table 3.2 To for Site Manage

(*Contin*

TOOL	DESCRIPTION	REQU	WEB SITE ADDRE
MKS Web Integrity	MKS' Web Integrity Web object management system for Web site maintenance and management; uses Web-server-based repository and a Java client interface; revision control, publishing control and security, audit trails.	Works with Win95 Netsca or MS IE, and several Netsca and Micros Web servers on NT and Unix.	www.mk
MS Visual Source Safe	Microsoft's version 5.0 of its software version control tool; includes site change management, Web link checker, site mapping; can integrate with MS FrontPage.	For Win3.1 and 95. NT, DOS, Unix, and Mac.	http:// msdn. microsof com/ssaf
HAHTSite debugging	Haht Software's integrated Web site development and management environment, for site authoring, publishing control, team development, link and object management, debugging.	Web site.	www.hah com
Mortar	Web site development, authoring, and validation tool; includes capabilities for multifile search/replace, custom tags, site mapping, and project management. Evaluation version available.	For Win95	www. bigpic.co
NetObject Team Fusion	Site authoring and management tool. Visual site structure editor, layout editor, graphics management, and staging/publishing control. Evaluation version available.	For NT.	www. netobject com

Summary

Web site management is one of the most important factors for creating a successful Web site. Your management team should work together to provide a functional Web site that will meet your business needs.

Chapter 4 will discuss risk management.

This page intentionally left blank.

CHAPTER 4
Risk Management

Web site risk management is a process that helps determine how an organization will be affected by exposure to risk on the Internet. Risk management can be used to minimize, control, or eliminate exposure to risks. IT managers can follow risk management procedures to gauge security concerns about an ecommerce site.

Risk management principles are key to defining policies and procedures with regard to keeping data secure by managing multiple levels of access controls for thousands of users. Applying risk management principles to data-security procedures will lend to implementing effective authentication and authorization processes throughout the network and within the Web application.

Risk management is inevitable for all Web development sites. There are two kinds of risks that are examined when evaluating a project: *opportunity risk,* which is the loss from avoiding risk, and *failure risk,* which is the loss from taking a risk but failing to achieve the expected goal. Loss may be financial, due to the downtime from a Web server, or it may be competitiveness in the Web

market. It may even be due to the development and acquisition of reusable software components or other valuable aspects of the Web site.

Managing risks requires that you as the tester or project manager set up clear guidelines of how the risks should be documented and tracked. These guidelines should be a work in progress; the individuals who are responsible for the risk management assessment should be able to access and update them as needed. Risk management can be addressed throughout the Web planning phase. You need to think of the risks before testing, during testing, after testing, and then again when the Web site is actually deployed. Following are several risk factors:

Probability. Probability is one risk method used to determine the likelihood of the

occurrence of a particular risk. The probabilities of risk are categorized as very low, low, medium, high, or very high. For example, server issues may be examined for their level of risk to the Web site. If the server goes down, it may have serious impact, which would make the risk very high.

Impact. Impact is used to determine the effect a risk would have on the project and how to handle the estimate of risk. Impact can be determined by categorizing risk as to whether they are negligible, critical, or catastrophic.

Overall risk. Overall risk is the risk to the project. The overall risk to the project can be determined by using estimates of risk probabilities and impacts. In calculating the overall risk, consider how this risk may affect other risks on the project, and make a note of them. A matrix can be used to determine the overall risk for each of effort, performance, and schedule (see Table 4.1).

Table 4.1 Risk Matrix

Impact/Probability	Very high	High	Medium	Low	Very low
Serious	High	High	Moderate	Moderate	Low
Critical	High	High	Moderate	Low	None
Negligible	Moderate	Low	Low	None	None

Planning for Risks

A plan should be developed to address each risk. A good rule of thumb for your risk toolkit is to ask the questions who, what, when, where, why, and how:

- *Who* is responsible for the risk management activity?
- *What* information is needed to track the status of the risk?
- *When* will resources be needed to perform the activity?
- *Where* will the resources be used?
- *Why* is the risk important?
- *How* will the risk plan be implemented?

A risk plan should be designed to determine tracking capability and the ability to maintain the risk management plan. There are two different types of risk planning. *Action planning* is

used to mitigate the risk by way of an immediate response. The probability of the risk occurring and/or the potential impact of the risk may be mitigated by dealing with the problem early in the project. This type of planning is considered proactive. *Contingency planning* (discussed in more detail in the section *Contingency Planning*) is used to monitor the risk and invoke a predetermined response. A trigger is set up, and if the trigger is reached, the contingency plan is put into effect. Following are several concerns in planning for risks:

Anticipate risks. When you are testing the Web site, you should have some preconceived idea of what part of the application may cause you problems. An example is testing your Web site to see if it will generate the correct calculated results from the shopping basket.

Eliminate risks. Potential problems can be identified before the testing process because the developer and programmer can deal with those issues during unit testing. It is important to make sure that the hardware you are using will work with the software before you begin any testing. Having a checklist of items of this type will ensure that you have everything in place before you start testing.

Reduce impact of risk. You can do several things to reduce the impact of risk. It is important to make sure you know everything there is to know about the Web site and previous releases of the Web site project. To lower risks for your Web site project make sure the testing team understands the basic components of the site and how the testing process should progress. It is also important to make sure that unit testing is being done after each phase of the coding process. Make sure you put into effect a complete test plan and document each phase of the software development. The testers should have prewritten scripts of the anticipated outcome of the test to follow.

Stay in control when things do not go as expected. As you test your Web site, expect that something will go wrong. Do not panic; instead, take control of the process and anticipate the next course of action as it pertains to the Web test process. Set up an analysis of the Web testing process and revise and rerun anything that did not go according to plan.

The best defense against certain key types of risks is to prepare a contingency and tracking plan that can be used to process and update your plan. Following are some important types of risks that a successful Web site should plan for:

- Failure risks
- Unclear mission statement
- No sponsor
- Project specifications are vague
- Project completion is one big chunk
- Inexperienced personnel

- Unavailable resources
- No project plan
- Unattainable schedule
- Expectations are unrealistic

Calculating Risks

Calculating and managing risks is a complex task. It is necessary to anticipate inside and outside factors that can affect the Web site. Some factors you may be able to control (lower risk); others you may find difficult to control (higher risk). There are several different risk categories:

Option risks. Cost versus benefits and return on investment (ROI) on each option. The ROI for a project is the ratio of the cost of resources required to the benefit generated by the project.

Opportunity risks. Risk of not doing the project.

Outsourcing risks. Having someone perform the project versus you doing the project.

Human risks. Can individuals successfully perform the process?

Technology risks. Do you have the adequate resources to perform the task?

Specific Risks

The infrastructure and the method of transmission are essential risks to consider when setting up, testing, and managing the site. Each item in the following list is subject to failure and should therefore be considered in risk management. You need to have a clear contingency plan to back up each component of this list. You should consider upgrades, revisions, new releases, new versions, and items that may become obsolete or subject to failure, including the following:

- Operating system
- Hardware
- Software
- Browser

-

- Internet service provider (ISP)
- Server
- Client
- Login scripts
- Error logs
- Hypertext Transfer Protocol (HTTP)
- Transmission Control Protocol/Internet Protocol (TCP/IP)
- URL

Controlling the Risk Process

A *risk process* identifies, analyzes, plans, and tracks risks. This process should be an important concern for the tester and should be in place before the project starts. If the process is not adequate, and risks are getting out of control, a new process may need to be formulated.

One way to look at risks is to see how the environment lends to potential failures, as illustrated in Figure 4.1.

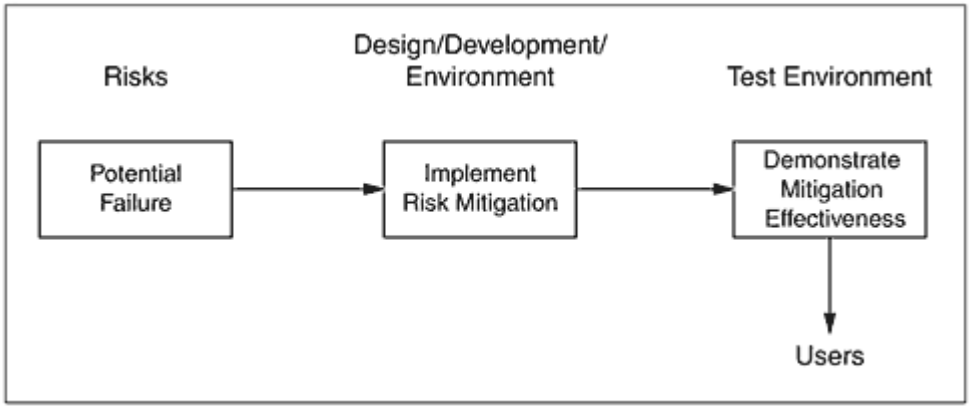

Figure 4.1 Illustration of the risk process.

The test processes are set up to address inherent risks. It is feasible to manage risks by organizing the tests around functions, requirements, structural components, or even a set of predefined tests that never change. Risk-based testing focuses on the effort of organizing the testing process for the Web site. If you are responsible for testing a Web site for which the impact of failure is extremely high, you may want to use a rigorous form of risk analysis. Such methods apply statistical models and/or comprehensively analyze hazards and failure modes.

A proactive way to manage risks is to apply certain procedures to handling risks, such as the following:

- Identify the possible risks that are involved in setting up the Web site.
- Make an assessment of the level of risk involved.
- Design a risk management plan based on the level of risks.
- Design a method to track the risks.
- Make the risk plan accessible to pertinent individuals to update.

Tracking Risks

Tracking risks is essential to the risk management process; if triggers go off, the entire team needs to be informed so that contingency plans can go into effect. Tracking is also useful as the project comes to the end of its development phase. Past knowledge may increase the chances of risk prevention and improvement in future projects. Resources are important as a part of the risk tracking process. For example, a simple shell script may be used to set up a test to check for user names and IDs. This type of test can be useful for determining the risk of the user logins.

```
# ! /bin/sh
#This script displays the date, time username and current directory
echo ∀Date and time is:∀
date
echo
echo ∀Your username is:  'whoami' \\n∀
echo ∀Your current directory  is: \\c∀
pwd
```


Tracking risks will enable you to identify risks and to follow through on the likelihood that the risks will occur on your Web site. Risks can be tracked by creating a tracking document. Each member of the team should submit a risk document for his or her particular responsibility. Following is an example of what should be included in the risk document:

- Name of risk
- Description of the risk
- Steps involved that would cause the risk to happen
- Results
- Probability of the risk
- Resources affected
- Comments
- Related risks

- Alternate plan

Risk Analysis

There are different ways to monitor how you would like to handle risks. Following are different methods that will help you analyze and address your risks:

Decide on the specific component of the Web that appears to have a high risk. Will you be looking at the entire Web site, a single component, or even a list of components?

Determine the severity of concern. Use a scale of normal, high, and low to rank the severity. Everything is presumed to be a normal risk unless there is reason for an assessment of a higher or a lower risk. Selecting a scale of

concern that is meaningful to your business is critical in the assessment of the level of severity.

Make individual input from your team key in identifying and foreseeing risks. Understanding the situation in which the Web site is set will help in developing a risk assessment. The team members will determine the different levels of risk that they foresee happening with their part of the Web site project.

After each risk is identified, decide on the importance of the risk and its severity. For each area of development a decision should be made to determine whether there would be risks in this particular area of development. You should then determine the level of severity. Record how you think this will affect your risk assessment. Determine how this type of risk is critical to the advancement of the Web site project.

Set up a plan that will be able to handle other risks as they occur. There will surely be risks that you may not even know about. It is critical to be able to deal with the uncertainties as well as the planned or foreseeable risks.

Record unknowns that will affect your ability to analyze the risk. During the process, you may feel that you are not able to assess the risk probability. If a certain portion of the Web site is complex, you may be unable to determine the type of risk involved. A risk is anything that may have a negative impact on your business or the performance of your business. As you progress through the risk analysis phase, it helps to make a list of risk items that are critical to your business.

Double-check the risk distribution. It's common to end up with a list of risks in which everything is considered to be equally risky. That may indeed be the case. On the other hand, it may be that your distribution of concerns is skewed because you're not willing to make tough choices about what to test and what not to test. Once you end up with a list of distributed risks, it is important to make sure you double-check them by taking a few examples of equal risks and asking whether those risks really are equal. Take some examples of risks that differ in magnitude and ask if it really does make sense to spend more time testing the higher risk and less time testing the lower risk. Confirm that the

distribution of risk magnitudes is correct.

Contingency Planning

As you will see, contingency planning is a vital part of software development. You can track all the items that are discussed in this chapter using a form like the one illustrated in Figure 4.2.

Specific details of your business' contingency management plans must be worked out by your users, IT personnel, all computer centers, and networking support teams. All contingency plans should address the following areas:

- Objectives of the plan (for example, continue normal operations, continue in a degraded mode, abort the function as quickly and as safely as possible, and so on).
- Criteria for invoking the plan (for example, missing a renovation milestone, reaching the projected Y2K-related failure date, experiencing serious system failures, and so on).

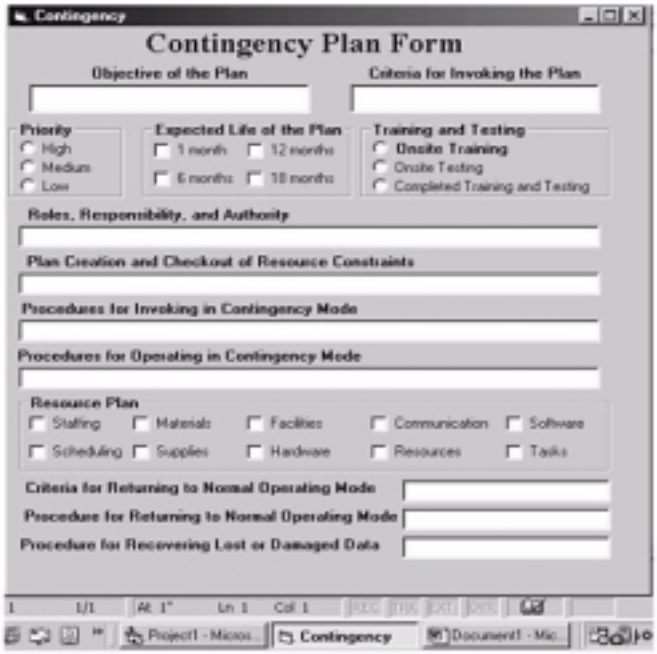

Figure 4.2 Example form used to track the elements of a contingency plan.

- Expected life of the plan. (How long can operations continue in contingency operating mode?)
- Roles, responsibilities, and authority.
- Plan(s) creation and checkout of resource constraints to plan for each contingency and

objective.

- Training on and testing of plans.
- Procedures for invoking contingency mode.
- Procedures for operating in contingency mode.
- Resource plan for operating in contingency mode (for example, staffing, scheduling, materials, supplies, facilities, temporary hardware and software, communications, and so on).
- Criteria for returning to normal operating mode.
- Procedures for returning to normal operating mode.
- Procedures for recovering lost or damaged data.

Because risks are anticipated, most risks can be planned for and addressed in the contingency plan. The risks can be documented and a proactive plan can be developed to deal with the risks.

As you go through Tables 4.2 and 4.3, think of the specific kinds of risks for which you would make a contingency or risk plan. Each area can cause a certain amount of risk for the other components of your Web site.

Version Control

Microsoft's Visual SafeSource is a tool that can be used to help reduce risk when moving an application from one version to the next. This tool can even be used for the documentation and version control of your Web site. The Visual SourceSafe 6.0 version control system is used for managing software and Web site development. It can be integrated with the Visual Basic, Visual C + +, Visual J++, Visual InterDev, Visual FoxPro development environments, and Microsoft Office applications. Visual SourceSafe will work with any file that is produced by any development language, authoring tool, or application.

Table 4.2 Web Site Testing Risk Areas

ARI	WH TO TES	IMPORTANCE	RISK
HTN	Link	You must be able to navigate a site through its links; the links are like the road map for the Web site.	Broken links can keep your site from functioning. If links are not updated, they may cause problems in navigating your site.

HTM	Spel	Because your Web site is your way of communicating, it is imperative that the spelling and grammar are constantly checked.	If you have important information on your site and it is misplaced or not understandable, people will not use the Web site.
HTM	Meta tags	By controlling your meta tags, you can control what information is placed into search engine database.	If your Web site does not have a connection to any search engines, people will not be aware that you have a site.
HTM	Title	Choosing an eye-catching Web site title is an important factor in the success of your Web site. You want a title that people will click on when they do a search.	If you have a bland Web site title or a name that is not familiar, you will have a hard time marketing yourself through search engines.
Acti	Forn	A form can be very complex; it must be tested constantly for integrity and comprehension.	If customers are not provided feedback after filling out a form, they will consider your Web site unreliable and will not use it.
Acti	Scrij	JavaScripts and VBScripts will give some programmers the shudders because they can be unreliable and cause browsers errors. It is one aspect of testing that must be taken seriously.	JavaScript and VBScript have their own unique characteristics. If your site is not tested for specific errors in scripting languages, people will encounter many errors and will not continue to use your site.
Acti	XMI	This must be well formed.	If the document is not well formed, you may find errors that could cause a lack of validity on your site.

Table 4.2 (*Continued*)

ARI	WHA TO TEST	IMPORTANCE	RISK
Acti	Inspe	Each site should have inspectors who constantly check for errors and validity.	If a site is not inspected, there may be errors that may cause your site to function improperly.
Site	Doma names	Your domain name must be correct. A domain name is considered a valuable piece of your Web site.	Many businesses rely on the strength of their domain name; if the information is incorrect, traffic cannot come into your site.
Site	Usabi	The site should be efficient, attractive, and practical to navigate.	If your Web site is not easy to use, customers will find alternative sites.

Site	Robo exclu	Exclude sensitive material from the site such as internal phone numbers.	If you have sensitive material that is not for the public viewing, it is imperative to exclude that information from your site. This could become a legal issue.
Site	Perfo	The site must be based on an understanding of the Web infrastructure.	If the infrastructure is not addressed properly, the site will have problems through deployment and Web hosting.
Site	Sourc contr	Understanding the sources of information is imperative to a successful Web site.	For quality assurance the tester must understand the use of forms, scripts, databases, and any source on the site that is used to gather information.
WW	Searc positi	Set up the site so that search engines will produce information and links to your site.	If search engines do not pick up key words for your business, it will be difficult to increase productivity and sales through the site.
WW	Serve logs	Analyze error logs for unexpected data.	If there are difficulties with specific issues on your site, you must be willing to use the logs to remedy all errors to keep the integrity and usage of your site intact.

Table 4.3 Busi

Requirem Risks

ITEM	MANAGEMENT	EXAMPLE
Personnel shortcoming	Top talent, team building, morale, cross training.	Hire a certified software engineer.
Unrealistic schedule and budget	Detailed multisource cost and schedule estimation, design to cost, incremental development, reuse, adjust requirements.	Use a project tracking tool, build one section of your Web site at a time.
Develop the wrong function	Organization analysis, mission analysis, user survey.	Early user input.

Develop wrong user interface	Task analysis, prototyping, scenarios, user characterization.	Talk to the user about the function, style, and expected workload.
Gold plating	Adjust requirements, prototyping, cost benefit analysis.	Design the project to the cost; if you want to add to or exceed the Web project, you must talk with management.
Continuing requirement changes	High change threshold, information hiding, defer changes.	Review the entire business requirement before starting the project.
Problems with externally furnished components	Benchmarking, inspection, reference checking, compatibility analysis.	Read the white papers for all the hardware and software to make sure all components will work together.
Problems with externally performed tasks	Reference checking, pre-award audits, competitive design or prototyping team building.	Check the references of everyone involved in the project.
Real-time performance shortfalls	Simulation, bench-marking, modeling, tuning, prototyping.	Unit testing is important to ensure that when a Web site is in real time it is functioning properly.
Training capabilities	Understand business requirements, provide trainers.	Training will enable end users to use the Web site and its functions.

Users can work at the file and project level while promoting file reuse because the project-oriented features of Visual SourceSafe make managing the day-to-day tasks associated with team-based application and Web site development more efficient. Visual SourceSafe is secure and scalable because it stores files in a secure repository maintained by an easy-to-use security system. It is a way to secure the code for your Web site project. The files stay protected but are still accessible to authorized users.

On the Web level, it will allow file sharing and will change tracking across projects. It also frees users from having to track which files are changed, by whom, when, and where. The result is increased team productivity for Web and PC file management.

One of the largest risks you will encounter when changing from one version of a product to another is the integrity of the existing version; this risk is lessened using Visual

SafeSource, which eliminates the need to rewrite previous code. If you would like a repository for your files, Visual SafeSource is a good tool to use.

Summary

When developing a Web site, you want to eliminate as many risks as possible. To do so you must understand the project and its business requirements. By anticipating risks you will be in a position to attack and rectify them. By using tables and charts and keeping ongoing documentation, you will be ready to handle the risk elements in setting up and testing a Web site.

Chapter 5 will discuss Web site testing tools.

This page intentionally left blank.

PART Two Web Testing Tools and Techniques

This page intentionally left blank.

CHAPTER 5 Web Site Testing Tools

In today's fast-paced world of Web site testing, automated testing tools play a crucial role in measuring, predicting, and controlling the application performance. There is a strong correlation between the use of testing tools and the ability for the site to scale the tools for accurate Web testing. Using testing tools can improve the performance of testing. It is best to combine manual and automated tools to achieve a good test of a product or application. Performance tests such as testing a system with thousands of users entering multiple types of transactions in many different types of combinations under varied conditions will determine the application's limitations within its Web site environment. Automated tools will help with systematically planning hardware purchases, isolating performance problems, and removing the human factor. This will enable testers to balance the test coverage across the entire

application.

The key is to learn how to evaluate the tools to find those that will most greatly benefit your business. A Web testing tool should be evaluated based on the objectives that you have set up in your Web testing plan. Tool selection should be based on criteria that are defined in your business requirements and testing objectives. In today's Web environment there are many different types of testing tools that you need to consider. The tool should have multiple functioning so that when you

are done with your current project, you can use the tool for other projects. The tool should also match your business objectives and goals. When selecting a tool, several things will help you through the evaluation process:

- Assign someone to evaluate the tools available.
- Define your business requirement criteria.
- Prepare a checklist that will help in the evaluation.
- Request a demonstration from the company.
- Evaluate all of the alternatives.
- Select the package.
- Install the test tool and train your team.

There are many different types of test tools for many different types of testing scenarios. Among them are GUI testing tools, load and stress testing, Internet and Web testing, requirements management, defect tracking, and telecommunication tools. Because the focus of this book is on Web testing, we will concentrate our efforts on understanding how to select the Web testing tool that best fits your needs. Keep in mind that Web application testing can be difficult and time consuming; therefore, automating the process will be beneficial. Also learning how to set up your Web testing effort is critical for the success of your project.

There are several factors to keep in mind when selecting a Web testing tool, including:

- Web volume
- Web architecture
- Web testing—automated or manual
- Web performance

Remember, you will not be relying entirely on this tool to solve all of your testing needs. The testing tool is meant to assist you in your testing needs. It is beneficial to also have a method of storing, maintaining, and reusing your test scripts. A good testing practice is to set

up a database of detailed test cases that you will be automating; this may also be used for a backup. Keeping these test

Test Script						
Test Script Number:					Priority: High	
System Tested:					Page:	
Test Case Number:				Date:	Tester:	
✓	Step	Action	Data Entry	Expected Results	Actual Results	Test Log ID

Figure 5.1 An example of a test script.

cases in a database will also assist you in the grouping of test cases, which will later be used in a test suite.

Writing a test script is just like writing a small program. Web test scripting can be automated by using a tool; manually setting up the steps involves working through a business requirement. Figure 5.1 is an example of a simple test script template. This script template captures the information you need to run a test. This type of script is the basis for information that is recorded when using an automated test tool. The difference is that the automated approach to scripting is generally begun by using some sort of recorder with the automated tool. Keep in mind that the recorder portion of the scripting tool is just a starting point; you should examine the script and modify it to meet your specific business requirements. Most automated testing tools have a scripting language that can be used to modify the script. For example, WinRunner's scripting language is TSL, which is very similar to a C language. Automated tools must be used by trained individuals. Some vendors of test tools offer training in their tools, which should be a factor when selecting an automated testing tool.

Types of Tools

Quite a selection of Web testing tools are available. This chapter provides a list of some of them, but keep in mind that this is just a sampling and there are many others you may want to consider. Tables 5.1 through 5.9 list the type of tool, a brief description, software/hardware requirements, and the URL.

Table 5.1 We Load Test Tools

TOO	DESCRIPTION	REQUIREMENT	URL
WebS	Load testing tool; includes link testing capabilities; can simulate up to 1,000 clients from a single IP address; also supports multiple IP addresses with or without aliases.	For Win98/2000/ NT4.0.	www. rehillnetw com
TestV	Collection of Web test tools for capture/playback, load testing, etc., from Software Research, Inc. Includes their XVirtual load generation tool.	For Win95/NT and Unix platforms.	www. soft.com/ products/ web/index html
WebF Train	Load test tool emphasizing ease of use. Supports all browsers and Web servers; simulates up to 200 users per playback machine at various connection speeds; records and allows viewing of exact bytes flowing between browser and server. Modem simulation allows each virtual user to be bandwidth limited. Can automatically handle variations in session-specific items such as cookies, usernames, passwords, and any other parameter to simulate multiple virtual users.	For NT, Linux, Solaris, most Unix variants.	www. webperfce com
WebS WebC	Load testing and capture/playback tools from Technovations. WebSizr load testing tool supports authentication, cookies, and redirects.	Requires Win95 or NT; WebCorder is a Java-based tool requiring JDK 1.0.2 or higher.	www. technovati com
Bencl Facto	Ecommerce load testing tool from Client/Server Solutions, Inc. Includes record/playback, Web form processing, user sessions, scripting, cookies, SSL. Also	Controlled by a Visual Control Center on NT.	www. benchmarl factory.co

Table 5.1 (*Conti*

TOC	DESCRIPTION	REQUIR	URL

Benc Fact (con	includes predeveloped industry standard benchmarks such as AS3AP, Set-Query, Wisconsin, WebStone, and others. Includes optimized database drivers for vendor-neutral comparisons of MS SQL Server, Oracle 7 and 8, Sybase System 11, ODBC, IBM's DB2 CLI, and Informix. Controlled by a Visual Control Center on NT.		
MS Web App Stres	Microsoft stress test tool created by Microsoft's Internal Tools Group (ITG) and subsequently made available for external use. Includes record/playback, script recording from browser, SSL, adjustable delay between requests, custom header per request, configurable number of threads, sockets, users, scripts, supports ASP, cookies. Site includes a useful Web load and stress testing tutorial. Appears to be freeware at present.	For testing ASP Web sites running on NT Server or Win2000, MSIE 4.0 or newer required.	www. homer.rte. microsoft. com
FOR	Load testing tools from Facilita Software for Web, client-server, network, and database systems.	Unix platforms.	www. facilita.co.uk/
Zeus Free Web Load Test Tool	Free Web benchmarking and load testing tool available as source code; will compile on any Unix platform. Similar to the tool that ships with Apache 1.3.x Web server.	Unix platform.	www. webper.zeus. co.uk/intro. html

continues

Table 5.1 Web Load Test Tools (*Continued*)

TOOL	DESCRIPTION	REQUIRE	URL
VeloM	Java-based Web load test tool; includes source code.	Requires JDK 1.1.6 or greater.	www.binevolve.co
http-L	Free load test application to generate Web server loads, from ACME Software.	For Unix.	www.acme.com/software/http_load
Micros WCAT Load Test Tool	Web load test tool from Microsoft for load testing of MS IIS.	Windows NT.	www.msdn.microsoft.co workshop/server/toolbo wcat.asp
Porten Web Load Test Tool	Loadtesting.com's low-priced Web load testing tool.	Written in Java, multi-platfo	www.loadtesting.c
WebA	WebART, from OCLC, Inc. load tests up to 100–200 simulated users; also includes functional and regression testing capabilities and capture/playback and scripting language.	For Win3.1/95/NT.	www.oclc.org/webart
WebL	Web load testing tool from Computer Associates. (This tool was originally Radviews WebLoad tool and it was distributed by Platinum Technologies, which is now a part of Computer Associates.)	For Win95/NT, Solaris, AIX.	www.ca.com/products/platinum/appdev/fe_iltps.htm/
Radvie WebL	WebLoad Web site load testing tool from Radview Software. Supports recording of Secure Sockets Layer (SSL) sessions, cookies, proxies, password authentication, dynamic HTML, multiple platforms.	For Win95/NT, Solaris, AIX.	www.radview.com

Table 5.2 Java Test Tools

TOOL	DESCRIPTION	REQUIR	URL
Panoram for Java	Contains six integrated Java tools; JavaSQA for object-oriented software quality measurement; JavaDocGen for Java code static analysis; JavaStructure for Java code structure analysis and diagramming; JavaDiagrammer for Java code logic analysis, control flow analysis and diagramming; JavaTest for test coverage analysis and test case minimization, etc.; and JavaPlayback for GUI operation capture and automatic playback.	For Win95/ 98/NT.	www. software automation. com/
McCabe Visual Test	McCabe Visual Testing ToolSet and McCabe Quality ToolSet, with Java coverage and metrics capabilities.	Java.	www. mccabe.com
jTest!	ParaSoft's automated white box Java test tool.	Java.	www. parasoft.com/ index.htm
AppletLc	Part of Radview's WebLoad Tool Set; for performance testing of Applets and Java-implemented protocols.	Radview' WebLoac Tool Set.	www. radview.com/
AssertMa	RST's code assertion toolkit for Java programmers and class level testers.	Java.	www. rstcorp.com

Table 5.3 Link Checking Tools

TOC	DESCRIPTION	REQUIREMEN	URL
Link Onli	LinkGuard Online, free on-the-Web broken link checking service.	Internet service.	www. linkguard.com

continues

Table 5.3 Link Checking Tools (*Continue*

TOC	DESCRIPTION	REQ	URL
Xenu Link Sleu	Freeware link checker by Tilman Hausherr; supports SSL Web sites; partial testing of FTP and gopher sites; detects and reports redirected URL; Site map.	For Win9	http://home.snafu tilman/xenulink.html
Link	Low-cost on-the-Web link checker; free trial period available. Automatically scheduled reporting by email.	Intern servic	www.linkalarm.c
Thes	Link checker for Mac; evaluation version available.	MacI	www.matterform com/
Aler Link	Link check tool.	For Win9 98/N	www.alertbookm download.h
I-Co Web	Link checker.	For Win9	www.futurearts.c ICONtrol.h
Riad	Link checker; evaluation copy available.	For Win9	www.riada.com

Link	Tetranet's tool for checking links, missing page titles, tetranet-slow pages, stale content, site mapping; from local drive or remote server.	For Win9	www. tetranet- software.co products/ linkbot.htm
Infol	Link checker program from BiggByte Software; can be automatically scheduled; includes FTP link checking; multiple page list and site list capabilities; customizable reports; changed-link checking; results can be exported to database. Freeware and evaluation versions available.	For Win9	www. biggbyte.co

Table 5.3 (*Co*

TOOL	DESCRIPTION	REQUIF	URL
LinkSca	Electronic Software Publishing Co.'s link checker/site mapping tool; capabilities include automated retesting of problem links, randomized order checking; can check for bad links caused by specified problems such as server not found, unauthorized access, doc not found, relocations, timeouts. Includes capabilities for central management of large multiple intranet/Internet sites.	For Unix, and Win98/N servers. Requires Perl 5.	www. elsop.com
CyberSp Link Test	Shareware link checker by Aman Software; capabilities include specified URL exclusions, ID/password entries, test resumption at interruption point, page size analysis, "what's new" reporting.	For Win3.1/ 95/NT.	www. cyberspyd com/cslnk html

Table 5.4 HTM

Validators

TOC	DESCRIPTION	REQUIREME	URL
Real	Shareware HTML validator based on SGML parser by Liam Quinn. Unicode-enabled, supports documents in virtually any language; supports HTML 4.0, HTML 3.2, HTML 3.0, and HTML 2.0; extensible–add proprietary HTML Document Type Definitions (DTDs) or change the existing ones;	For Win95/ 98/NT; MSIE 4.0 or greater, HTML Help 1.1.	http:// arealvalidato com/

continues

Table 5.4 HTML Validators (*Continued*)

TOOL	DESCRIPTION	REQUIR	URL
RealValidator (cont.)	fetches external DTDs by HTTP and caches them for faster validation; HTML 3.2 and HTML 4.0 references included as HTML Help.		
CSE 3310 HTML Validator	Shareware HTML validator.	For Win95/NT	www. htmlvalidator. com/

Table 5.5 Free HTML Validators and Link Checkers

TOO	DESCRIPTION	REQUIREMENT	URL
WDG HTM Valid	Web Design Group's validator; latest HTML version support, flexible input methods, user-friendly error messages.	Web site.	www. htmlhelp. com/tools/ validator/
Metal	Northern Webs free on-the-Web meta tag checker; professional version available for purchase also.	Web site.	http:// northern webs.com/ set/setsimjr. html
Web Page Purifi	Free on-the-Web HTML checker allows viewing a page purified to HTML 2.0, HTML 3.2, HTML 4.0, or WebTV 1.1. standards.	Web site.	www. delorie.com/ web/ purify.html
W3C HTM Valid Servi	HTML validation site run by the WWW Consortium (the folks who set Web standards); handles one URL at a time; site includes specifications info for several recent HTML standards.	Web site.	http:// validator. w3.org/
NetM	Link checker and HTML validator by Monte Carlo Software. Type in the site URL to check, then later receive email with the URL of a Web page that contains	Standard HTML versions or MSIE or Netscape versions or combinations.	www. netmechanic com

Table 5.5 (*Conti*

TOC	DESCRIPTION	REC	URL
NetM (con	the results. Validator can choose among standard HTML versions or MSIE or Netscape versions or combinations.		
Bobl	HTML validator; can validate against AOL browsers, MSIE, Netscape, WebTV, specific HTML standards, and Lynx. Can run validation against multiple standards simultaneously; also checks page for accessibility to users with disabilities.	Web site.	www.cast.org/bobby/
Doct HTM	Site with online Web page checker by Imagiware. Checks spelling, forms, tables, tag usage, link. Analyze by page or by site. Selectable depth for sites. Also has info about SiteDoctor product, which checks an entire site.	Web site.	www2.imagiware.com/RxHTML/index_nofram html
EWS Web Gate	Site with online HTML validator; somewhat configurable.	Web site.	www.cen.uiuc.edu/cgi-bin/weblint
Web Page Back Com Viev	On-the-Web HTML checker; will serve a Web page to you with various selectable tags switched on or off; very large selection of browser types; to check how various browsers or versions might see a page.	Web site.	www.delorie.com/web/wpbcv.html
Mind	NetMind's MindIt (formerly URL-minder) sends email whenever a URL is updated. Can enter many in a maintained list. Can be used to keep track of changes to external linked documents and pages.	Web site.	www.netmind.com/html/individual.html

Table 5.6 Log Analysis Tools

TOOL	DESCRIPTION	REQU	URL
HTTPD Log Analyzers list	Most extensive log analysis tool listing on the Web–more than 100 listed with short descriptions of each, organized by platform: Unix, Win, NT, Mac.	Unix, Win, NT, Mac.	www.uu.se/Software/Analyzers/Access-analyzers.html
Web Developers' Virtual Library Log Analyzer Listing	Smaller log analysis tools listing, includes about 30 tools with descriptions.	Web site.	www.stars.com/Vlib/Software/Statistics.html

Tab

5.7

Wel

Tes

Too

TOO	DESCRIPTION	REQ	URI
Cast IP Sock	Castalia IP socket tester; freeware available via Download.com's site in the Internet-> Tools and Utilities section; can test port-specific socket-based IP comms; logging capabilities.	For Win	www redh
Test			netw

Net.I	Performance monitoring tool from VitalSoft, division of Lucent Technologies. Assists in monitoring, isolating, and correcting bottlenecks, problems, and isolating problems to modem, ISP, backbone, or server. Can monitor connections' number of hops, download rates, server vs. network effects, maintain metrics over time such as server availability, connection rates, bytes transferred, etc. The Net.Medic Pro version is geared to Webmasters and ISPs, produces more detail, and can perform continuous monitoring.	Win	www ins.c softv med inde

Table 5.7 (*Contin*

TOO	DESCRIPTION	REQUIREMENT	URL
MRT	Multi Router Traffic Grapher–tool utilizing Simple Network Management Protocol (SNMP) to monitoring traffic loads on network links; generates reports as Web pages with (graphic interchange file) GIF graphics on inbound and outbound traffic.	For Unix and NT.	http://ee-staff. ethz.ch/ ~oetiker/ webtools/ mrtg/ mrtg.html/
SACc	Network administrative security tool from SAC Technologies–mouse-sized PC-compatible fingerprint reader device for integrating with network security systems. Can be used to control access to network and server management and security.	Security tool for the network.	www. sacman.com/
WebM	Web usability testing and evaluation tool suite from U.S. Govt. NIST. Source code available.	For Unix, Win95/NT.	http://zing. ncsl.nist.gov/ webmet/
WebI	Debugging tool from Aman Software for monitoring HTTP protocol; sends and receives; handles HTTP 0.9/1.0/1.1; allows for entry of custom headers.	HTTP 0.9/1.0/1.1.	www. cyberspyder. com/
Brow	Browser emulator from Codo Development that allows viewing of HTML in emulation's of Netscape, IE, and W3C standards.	For Win95/NT.	www. bizdomain.com browserola/

Enter Mind	NetMind's server-based application for keeping track of changing information on extranets/intranets/Internet. Allows each user to have granular control over what is tracked and update frequency.	Works with most NT or Solaris servers; supports secure SSL, Basic and NT LAN Manager authentication and proxy servers.	www. netmind.com/ html/ enterprise_ minder.html

Tab

5.8

Tes

Too

TOO	DESCRIPTION	REQUIR	URI
Mac	Functional and regression test tool for ebusiness Web sites from Watchfire.com. Capture/playback capabilities, includes spreadsheet-like data table feature that allows use of varying values for user accounts, order baskets, bank account transactions, credit card types, etc. Element testing feature allows testing for specific page elements including forms, graphics, text, and links. Test Plan feature allows automatic generation of test plans by generating plain English explanations of recording test scripts for tracking Web development and QA processes.	Requires NT4.0 SP4 or later, Win95/98 MSIE 4.01 SP1 or higher.	wwv watc

Table 5.9 W site Security Test Tools

TOOL	DESCRIPTION	REQUIRE	URL

HackerShield	Security scanner from BindView; examines servers, workstations, routers, hubs, printers, and any other devices with an IP address on a network, for security holes that hackers could use to break in. Scans any device on network regardless of platform; updates are sent via secure email and automatically integrated into database of security checks.	Runs under NT4.0 with SP 4 or better and MSIE.	www. bindview
Cyber Attack Defense System	Monitors and analyzes suspicious activity at Internet gateways and public Web servers; takes	For Solaris and NT.	www. checkpoi

Table 5.9 (*Continue*

TOC	DESCRIPTION	REQUIR	URL
Cybe Atta Defe Syst (con	action to stop cyber attacks including distributed Denial of Service attacks; integrates with third-party OPSEC products such as switches and load balancing solutions.		
ITS4	Open source software tool from RST Corp. automatically scans C/C + + source code for potential security vulnerabilities.	For Unix and Windows with CygWin.	www. rstcorp.com/its4
Host	Suite of security test and management tools from DMW Worldwide. Checks for presence of sniffers and insecure network configuration, secure file removal by repetitive overwriting of location, checks for pre-existing security problems against vulnerability database, file/directory permissions management and monitoring, password security testing, user management and account security checking, security reporting and test scheduling.	For Unix.	www. dmwworldwide. com/

Web Secu Anal	Web site tool to detect and fix security problems. Includes periodically updated Expert knowledge base.	For Win95/ 98/2000/	www. webtrends.com
Secu Scan	Cisco's product for detecting and reporting on Internet server and network vulnerabilities; risk management; network mapping.	For NT or Solaris.	www. cisco.com
Insp	Shavlik Technology's security analysis and reporting tool.	For NT.	www. shavlik.com

continues

Tab

5.9

Site

Sec

Tes

Too

(*Co*

TOC	DESCRIPTION	REQ	URI
Nete Site	Netect's product for analyzing internal and external security vulnerabilities.	For Solari	wwv nete

SAF	Collection of security products from Internet Security Systems for security assessment, intrusion detection, and security management. Includes Internet Scanner for security scans of devices at the network level; Database Scanner for protecting database applications through security policy creation, compliance, and enforcement; available for Oracle, MS SQL Server, and Sybase. System Scanner searches online operation to provide host-based security assessment targeting weaknesses undetectable through network scanning.	For NT and Unix platfo	www iss.n

As you can see, this is quite a list of tools for you to explore and evaluate. You should take your time and go through all the possible tools for your site before you make a decision on the test tool.

Define Your Business Requirement Criteria

A well-planned test automation project will make use of the business requirements for a total Web test process. Test automation is a definite programming effort for the tester. Every tool may have a recording process, but every test case

developed may need to be modified to specifically handle the test case's scenario for each business requirement. When suites of test cases are developed, you need to make sure that the cases are set up to run in a functional test suite. The suite should take into account the type of site you have and the components that need to be tested.

Organizing the test scripts according to the requirements can make the test suite functional. Setting up test cases involves using a data-driven method; this is also relevant for Web testing. For example, when testing a GUI on a Web site, care should be given to designing test scripts according to a logical structure. Because you are the one responsible for the tool selection, you should establish decision criteria that you want to use for the test. These criteria will enable you to buy from vendors who will give support and training as required. The correct operating system and its ease of use and organization are important components to consider. Mapping out the design of your test cases by working through the application will lend to setting up the test suites. This may involve screen shots and business requirements.

An important point is that automation enhances the manual aspect of testing. More complicated tentative testing can be applied to the dynamic areas. The expected result is that a smaller, yet more technically adept testing staff will be required in this type of testing environment.

Prepare a Checklist to Help in the Evaluation

To make a sound decision on a testing tool, you need to have a checklist that will help in the evaluation of the tool (see Table 5.10). The cost of the testing tool, usability, and the amount of time available to test tend to be the first critical components in the decision-making

process. Many organizations tend to not use a testing tool because of cost constraints.

Request a Demonstration from the Company

Companies are ready to sell their product, so having them set up a demonstration is beneficial to them as well as to you. The companies in the following

Tabl

5.10

Eval

Chec

ITEM	YES	NO	N/A	COM
1. Is the test tool easy to implement and use?				
2. Is training available from the vendor or other outside sources?				
3. Are employees available who have experience with the tools?				
4. Can a test environment be set up to use the tools?				
5. Can the tool be used for other areas of the testing process such as project management?				
6. Are there any white papers on the test tool so you can compare it to other tools?				
7. Is the cost included in the current testing budget?				
8. Is there a tool to evaluate the features of Web testing?				

sections represent the leaders of the test tool industry. Each of them will provide information and demonstrations if requested.

Segue

Segue (www.segue.com) supports Visual Basic, PowerBuilder, SQL Windows, Web

Applications, Java, and also mainframe and AS/400 applications via Terminal Emulation. It uses its own scripting language, 4GL, which is very much like C + +.

One of Segue's tools, SilkPerformer 4.0, optimizes ebusiness reliability. It accurately predicts the capacity and constraints of Web applications prior to their launch, optimizes their performance throughout their entire life cycle, and

provides insightful answers to performance challenges. This will speed their deployment, ensure scalability, and maximize uptime.

SilkPerformer works reliably with B2C, B2B, and wireless Web applications for hundreds of ebusinesses ranging from emerging dot-coms to Fortune 100 enterprises. SilkPerformer 4.0 is one of the tools that is being used to test complex Web applications under the heaviest of loads and bursts of activity. SilkPerformer 4.0 provides real-world simulations. It accurately emulates the most realistic ebusiness conditions by simulating a nearly infinite number of simultaneous users and traffic scenarios with a single script. It can also simulate multiple combinations of protocols and computing environments using a single recorder to capture and replay scripts.

End-to-end reliability. It lets you determine your site's scalability from the earliest stages of development right through final production.

Firewall support. It maintains firewall integrity while monitoring all application and database servers across any wide area network or Internet infrastructure.

Agent health control. To ensure valid test results, SilkPerformer continuously monitors the central processing unit (CPU) utilization, memory requirements, and responsiveness of each agent.

Mercury Interactive

Mercury Interactive (www.merc-int.com) tools support Visual Basic, PowerBuilder, SQL, Windows, SAP/R3, Oracle Developer 2000, Borland-Delphi, Web-based application (WebTest), Java, and all mainframe and AS/400 applications. Terminal emulation such as Attachmate can be used to set up the emulation for the mainframe. Mercury uses a C-like test scripting language called TSL. Mercury has the following tools:

TestDirector. Helps manage the testing process. This tool can create a database of manual and automated tests, build test cycles, and execute tests and report and track bugs.

LoadRunner. Has a Web load testing feature that supports HTTP, HTML, and Java applets. You can create scripts by recording the actions of a user or user groups surfing a Web site. You can also determine the maximum number of concurrent users a Web site can handle.

Astra. A visual Web site management tool that can scan the entire Web site, highlighting

functional areas with color-coded links and URLs to give a visual map of the site. This tool can pinpoint broken links and access problems.

ActiveTest. Can help ensure that users have a positive experience with a Web site. ActiveTest is a hosted, Web-based testing service that conducts full-scale stress testing of your Web site. By emulating the behavior of thousands of customers using your Web application, ActiveTest identifies bottlenecks and capacity constraints before they affect your customers. Mercury's site will allow you to try ActiveTest.

You can view the results with ActiveTest real-time online monitors and simultaneously discuss your findings with Mercury Interactive personnel. They can communicate with you via a conference call, providing answers to your questions and giving you helpful suggestions. After the test is complete, you will have access to online graphs and reports. As a hosted service, ActiveTest provides all the technology and hardware resources you need for Web performance management, as well as the expertise of Mercury's load testing specialists. You would not need to invest in hardware, software, and training. As a result, you can save money and focus on your core business. ActiveTest is powered by Mercury Interactive's LoadRunner, one of the industries leading standard load testing tool.

Rational

Rational tools (www.rational.com) support testing of 32- and 16-bit Windows objects and components, including OLE controls (OCXs), Internet ActiveX controls, Visual Basic controls (VBXs), Visual Basic objects, PowerBuilder objects,Win32 controls, and others. It also has its own scripting language.

Rational has a product called Rational SiteLoad, which the company says is an easy-to-use, scalable tool that simulates Internet traffic and provides developers with precise real-time information on site performance. According to their materials:

> Rational recognizes that in a fast-paced industry that demands high-quality results, testers often spend more time learning complex tools than performing the necessary

tests on their Web sites. To address these challenges Rational offers Rational SiteLoad, which includes the following benefits:

> **Speed.** In today's Internet environment, where speed is of the utmost importance, Rational SiteLoad delivers meaningful test results in just minutes.
>
> **Browser-based operation.** Unlike existing tools, Rational SiteLoad was designed for Web personnel rather than professional testers. Written in Java, Rational SiteLoad runs on a Web server and is accessed through a browser. This reduces the learning curve and allows a wider range of users to implement the testing solution.
>
> **Auto-ramp.** Rational SiteLoad allows users to incrementally increase the user load for a pre-set length of time or until a performance threshold is met. This saves time by executing multiple test runs simultaneously.
>
> **Scalability.** Rational SiteLoad is highly scalable and allows for the simulation of tests for ten to tens of thousands of users.
>
> **RealStatus.** Rational SiteLoad RealStatus reports provide real-time graphs that detail the response

times and resources used by the testing systems.

Resource monitor. The impact of the user load on system resources (e.g., disk space, memory) can be easily monitored, helping to eliminate bottlenecks.

Platform independent. As a Web application, Rational SiteLoad can be accessed from any platform, and can be hosted on Windows, Solaris or Linux.

AutoTester

AutoTester (www.autotester.com) is an automated testing solution for Windows 3.X, Windows 95, Windows NT, and OS/2 applications. AutoTester has a link check and site test tool that includes scripting. AutoTester also comes with the ability to test Java applets and ActiveX controls.

This test tool provides capture/replay test creation and can store the tests as documented and maintainable tests. The product also includes an easy menu-driven interface along with a powerful set of commands to customize scripts. The AutoTester Company offers quality assurance experts who can work with you on site to provide software training, implementation assistance, and project support. This tool can help in providing functional and regression testing techniques.

AutoTester deals with specific issues involved in functional and regression testing such as immediate testing productivity, ability to create simple-to-understand tests, self-documented business process tests, easy-to-read

reports, interface testing, and testing on host-legacy systems. AutoTester can generate a structured, well-documented test as follows:

- Captures your tests as a series of object- and data-aware events resulting in a highly flexible, reusable, and maintainable test library.
- Automatically adjusts during playback to accommodate changes in size or position of your application icons, objects, or controls, significantly reducing your maintenance effort.
- For SAP R/3 environments, AutoTester creates R/3 object-aware tests that intelligently interact with R/3 GUI objects at the user interface level.

With AutoTester, your organization can benefit from true unattended GUI testing. AutoTester tests are intelligent scripts complete with logic that duplicate your own expectations and decision points. For example, AutoTester:

- Provides on-the-fly verification of application object contents and state, displayed text values regardless of font, and complex bitmap images during test playback.
- Identifies unexpected application or system responses during execution and adjusts accordingly to continue testing.

- Logs the results of application failures and system errors and then continues the testing process using advanced recovery options.
- Restarts your application or operating system if necessary.
- Supports playback synchronization, which provides proper test playback regardless of system performance.

AutoTester automatically documents in plain English every step of every test you perform. Test case numbers and test requirement identifiers should include detailed descriptions for reference purposes. After test execution, detailed results such as the following are available online or in report format for immediate review and analysis:

- View summary and detail results of each test run online, in a customizable format, with the Results Viewer.
- Review complete test results with the Test Log, which details each test step performed and whether it passed or failed.
- Track application errors with the Error Log, which will document the details of each failed test.
- Monitor system performance of host and server-based applications through response times.

The AutoTester scripting language is a powerful command set that is used to supplement your captured tests or to develop tests before your application code is completed. The command set is presented with simple menus that step users through development with online help and familiar controls. Developers can use the scripting capabilities to incorporate conditional branching, looping, and external test case data into their tests.

Under Windows 3.X and Windows 95, AutoTester allows you to test not only your GUI-based applications but your character-based host applications as well. Through its ability to work with 16-bit 3270 and 5250 terminal emulators, AutoTester enables you to test distributed non-GUI applications running on remote platforms. Client system requirements are Windows NT or Windows 95, 8 megabytes minimum of memory plus operating system requirements, and 6 megabytes minimum of disk storage. AutoTester's technology, methodology, services, and support are the building blocks for an enterprisewide software quality assurance solution for your organization. AutoTester is unique in that it provides solutions addressing the entire spectrum of software quality assurance issues that organizations are facing today.

CompuWare

CompuWare (www.compuware.com) supports testing of Internet, client-server, and mainframe CICS, and VTAM applications will work directly on the mainframe as well as from a PC workstation (QAHiperstation+ and QAPlayback+). CompuWare's proven testing methodologies, experienced professional services resources, and industry-leading products

will help you thoroughly test your ecommerce applications and make sure they are production ready.

If you automate your testing process for ecommerce applications, CompuWare will become an absolute necessity for successful Web application testing. Ecommerce applications are more complex than the traditional, legacy application. Change is frequent on the Internet and can result in problems anywhere from the Web browser to your Web server, application server, network, or mainframe.

CompuWare's *QA*Center testing products offer automated testing solutions for ecommerce applications:

Reliability. To keep pace with the rest of the Internet, your Web site and other ecommerce investments are constantly changing and growing. *QA*Center can help you to establish a Web testing process that will prepare your ecommerce applications for the rigorous and unpredictable demands of the Web.

Integrity. As your Web site grows, so does the chance that your site may experience embarrassing and costly failures. Web site analysis, using *QA*Center, can ensure the integrity of your online investments.

Scalability and performance. Performance on demand is key to success on the Internet. *QA*Center tests your Web site and online applications in a production-like environment to ensure your site will scale and perform according to your expectations.

CompuWare File-AID products help you pull together test data from multiple sources to create, move, convert, reformat, subset, and validate your test data bed. The testing methodology has helped organizations test more efficiently and effectively.

Empirix

Empirix (www.empirix.com) specializes in ebusiness application testing software and was established with the goal of providing best-in-class testing products and expertise for business-critical Internet and intranet applications.

Empirix offers a full solution for automatically testing Web-based applications with its product e-TEST suite. RSW is known for delivering user-friendly tools

that are both quick to implement and easily adaptable to varied in-house testing resources (see Figure 5.2). Within a relatively short time frame, Empirix has had a substantial impact on the market for automated testing tools targeting Web environments and is optimized for multiple application servers such as Allaire's ColdFusion, ATG's Dynamo, BEA's WebLogic, IBM's WebSphere, and Microsoft's ASP.

Empirix's newest product, the e-TEST suite for the Wireless Application Protocol (WAP), includes added functionality to test wireless applications. Additionally in 2000, Empirix became the first to offer a specialized tool for testing the scalability and integrity of

middle-tier applications built with Enterprise JavaBeans (EJBs). The product EJB-test is optimized for leading application servers from BEA (WebLogic) and IBM (WebSphere) and generates multithreaded Java test clients to measure performance, scalability, and functionality of EJB component-based applications.

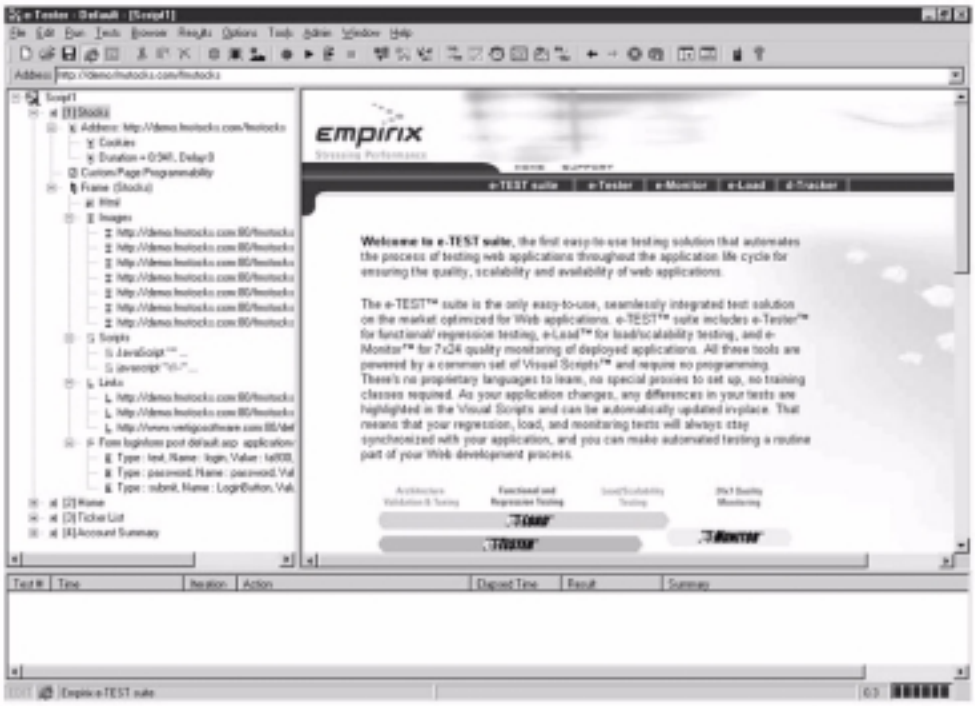

Figure 5.2 e-TEST.

Page 118

Parasoft

Parasoft provides for n-tier software applications with Web interfaces. Developers have been looking for the same supporting tools in Web development that they normally use in other types of development. WebKing is a unique tool that allows developers to prevent and detect errors as they build n-tier Web applications (see Figure 5.3).

WebKing takes testing techniques that have been proven to improve the quality of C/C + + and Java code and automatically applies them to dynamic Web applications. Web developers can use WebKing to automate white box, black box, and regression testing as well as Web box testing, which is a new method of performing unit testing on dynamic pages.

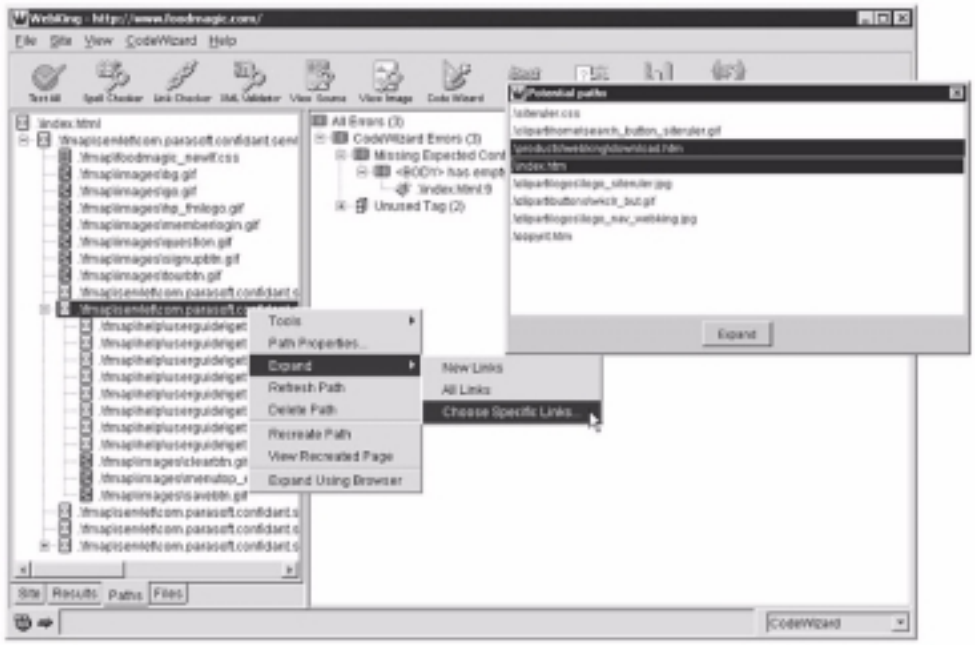

Figure 5.3 WebKing.

Page 119

Cyrano

Cyrano has a new Web test tool called WebTester v4.2. It is a complete Web solution that allows you to create, maintain, and execute regression and functional testing, load and scalability testing, and availability and reliability testing for your Web-based applications. New features and enhancements in WebTester v4.2 include:

Support for Web applications containing Java applets. Version 4.2 of the Cyrano WebTester suite allows users to quickly perform functional and load testing of Java applets, making Cyrano WebTester suite the only integrated testing solution for Web applications containing Java applets. With a single script, Cyrano WebTester now supports a broad range of technology, and that same script drives all facets of testing, including:

- Load
- Regression
- Monitoring

Using Cyrano WebTester suite, companies working in tightly compressed time frames on ebusiness projects will be able to experience immediate benefits.

Web-based reporting and new data sources. Cyrano Web Reporter has been extended with a Web-based interface. Now Web Reporter's extensive test reports correlating user response times and system performance statistics can be viewed from anywhere on the network via Cyrano WebTester suite's new Web Grapher. This means that anyone can produce and view Cyrano WebTester's extensive test reports and graphs, allowing

cross-functional groups to view, initiate, and share reports throughout the company. Development and deployment teams can quickly identify how their applications perform under load and in production and can easily detect where performance bottlenecks exist. Along with IIS and Apache, the latest version now supports Netscape Enterprise Server (NES), as well as a large variety of application servers, database servers, operating

systems, and network elements, giving users a broader range of systems supported to monitor server-side activity.

International language support. The Cyrano WebTester suite has been optimized to fully support all foreign languages, including Asian languages using double-byte character sets (Japanese, Chinese, Korean, Vietnamese), Eastern European languages, and others. As an ebusiness grows, the company can continue to use Cyrano's WebTester suite for expansion into international markets.

Conclusion

The tools listed are just a few of the many that are available. As you review, evaluate, and reach a decision about the tools that are best for your needs, keep the following in mind:

Evaluate all of the alternatives. There are many different tools available; that is why using the checklist and obtaining as many different evaluations for the tools is important. As you can see, there are many tools available. Make sure that you evaluate the tool for your specific needs. Remember to talk to several vendors before you make your final decision.

Select the package. The process of automating test scripts is expensive. The software vendors might have you think otherwise, but if you plan to reuse scripts from build to build and version to version, the test system must be designed properly and the scripts themselves may require serious modification after record and playback. On the other hand, test automation is an investment and can easily pay for itself given the right conditions.

Install the test tool and train your team. Installing the test tool and training the team is the last step before beginning the actual test. Setting up training programs and understanding and preparing the test cases, test scripts, and documentation are important for the success of the project. In-house training and training classes are available from vendors. It is important for the users and testers to become involved in the training process and the preparation of a training program.

Summary

As you can see from this chapter, there are many factors to consider in making a decision for setting up and running an automated testing tool. The key is to determine what the best tool is for your needs, train your staff, and conduct your test using the selected tool.

Chapter 6 will discuss preparing the Web environment for testing.

This page intentionally left blank.

CHAPTER 6
Preparing the Web Environment for Testing

As more and more applications become Web based, testing will become an increasingly larger issue. Testing was thought to be as simple as putting your URL into a browser; if the links worked, you were in business. As time went on and large corporate Web sites started to crash, costing millions of dollars in revenue, Web-based testing began to be taken seriously.

Web sites are considered a type of client-server system. It is important for the Web site to work on your desktop before it is transferred to the Internet. If the Web site is set up to run on your client-server system, there are certain questions that need to be answered about the system's environment. Examples are:

- What type of platform does the software run on behind the scenes?
- How many tiers are involved in the architecture of the system?
- Does the Web site use only static Web pages? If so, where do the pages reside?

-
- Does the Web site use Active Server Pages? If so, how are the pages published?
- Does your Web site publish pages on a schedule? If so, do the pages have macros that need to run on a software product such as Microsoft Word?
- Does your Web site have any image documenting behind it? If so, will the server running the imaging document work with your present server?
- Do you have information stored in a database? If so, will your database run off your server?
- If you are using a database such as SQL Server to run Active Server Pages do you have a true connection string set between your database and your Domain Name System (DNS).
- Can you view your pages in any browser? If not, do you have an alternative plan such

as text documents or the ability to create PDF files?

The answers to these questions will help identify how to set up your test environment.

Setting Up a Test Environment

Setting up the test environment for a Web-based application can be tricky. You must make sure that the engineers understand the specific application design and the environment for your site. They also must make sure that all software and hardware work with the automated tool you select. The environmental setup engineer should have the architecture plan devised for the implementation of your Web site.

Figure 6.1 is an example of a *proxy server*, which provides access to files from other servers by retrieving them from the local cache or from the remote server. This type of setup can be used for users who want to connect directly to Web sites. Examining this particular scheme will provide an engineer with an understanding of the networking architecture necessary to perform and maintain testing.

Figure 6.2 is an example of a proxy server that allows users to connect to Web sites by first going through the proxy server. This type of network may cause additional concerns because the proxy server needs to be set to run the tests from different servers.

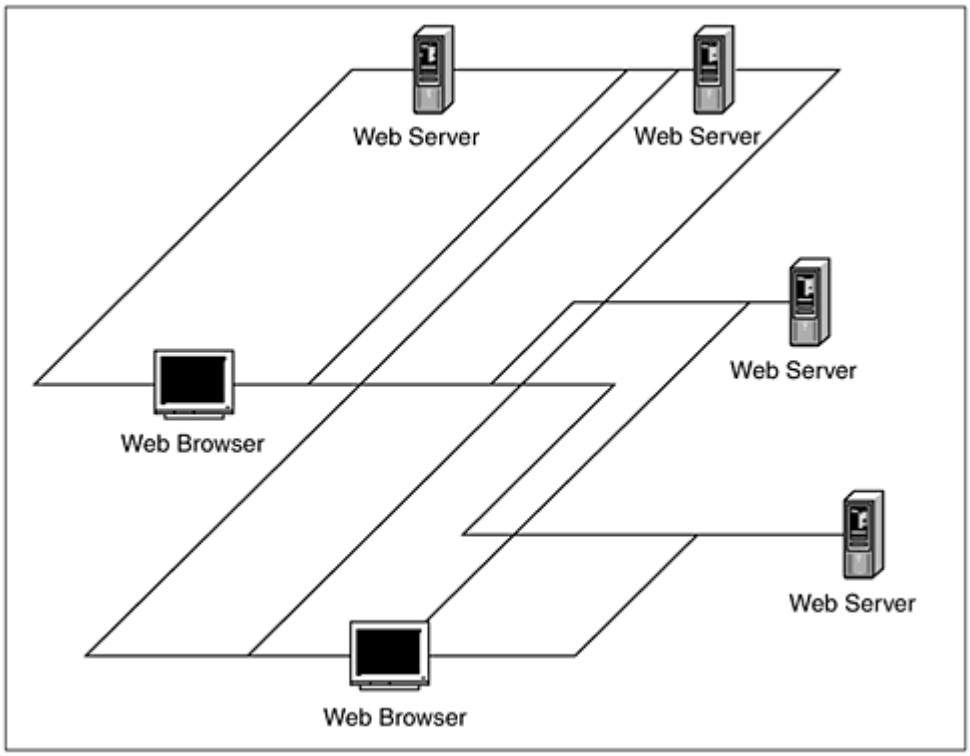

Figure 6.1 Example of a proxy server.

It is important to understand the distinctive layouts of different servers to better understand how the testing will work between the browser and those servers. The environment engineers need to be able to simulate the application to be tested, which is best achieved by understanding how the servers are designed since that will impact how the

application will be tested:

HyperText Markup Language (HTML). HTML is the language used to create Web pages, which includes hyperlinks and markup for text formatting.

TCP/IP communications. The Transmission Control Protocol (TCP) is on top of the Internet Protocol (IP). The Internet Protocol is a connectionless protocol, which provides packet routing. TCP is connection-oriented and provides reliable communication and multiplexing.

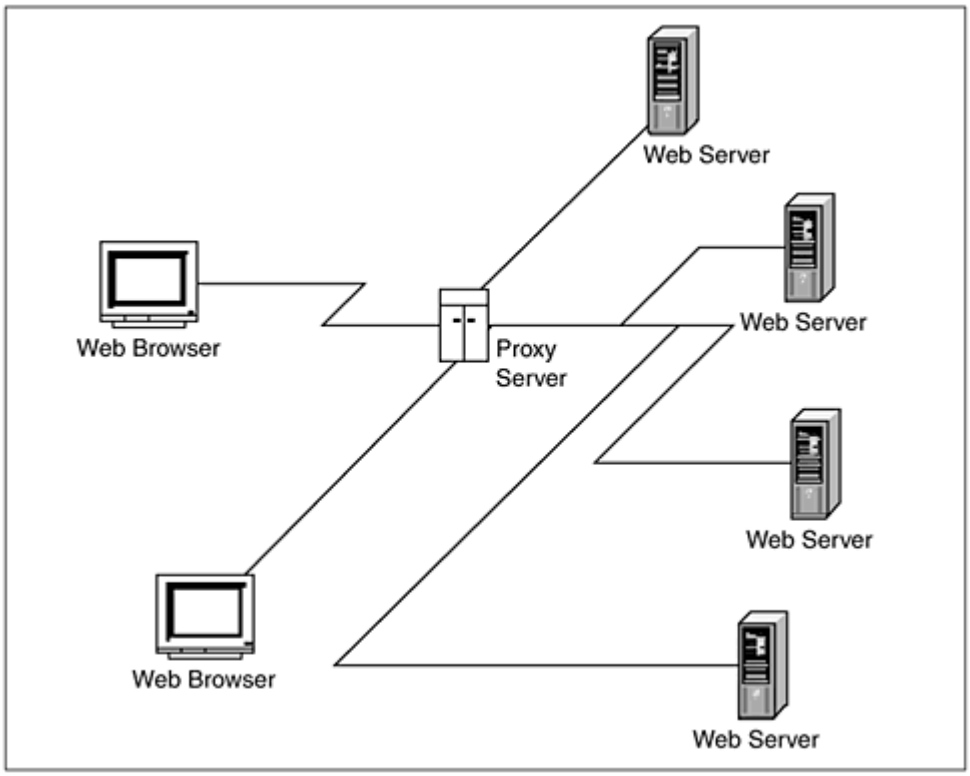

Figure 6.2 Proxy server for connection to Web sites.

Internet connections. Internet connections are ways to transmit data that requires a connection be established first. When the connection is established, the data is transferred and the connection is released. Examples of connection-oriented transmission are TCP and most wide area networks (WANs).

Firewall. A firewall is a hardware and/or software boundary that prevents unauthorized users from accessing restricted files on a network. The part of the network that is not behind the firewall is available to all users. There are three standard firewall architectures:

- Dual-host gateway
- Screened-host firewall system
- Demilitarized zone firewall

Applications and scripts that run on Web pages. These are accessory programs that enhance a main application, such as applets, JavaScript, or plug-in applications. An example is the set of additional tools and effects available to the Photoshop image editor in the plug-ins folder. There are many plug-ins for Web browsers, such as Shockwave and Crescendo MIDI player that will give the browser special capabilities, especially for multimedia Web sites. Another example is an applet, which is a utility. On the World Wide Web, many applets are written in Java language and attached to HTML documents.

Applications that run on the server side. Programs that run on the server side of the application can include CGI scripts, database interfaces, dynamic page generators, or ASP. For example, a CGI script is a program that runs on a Web server in response to input from a browser. The CGI script is the link between the server and a program running on the system, for example, a database. CGI scripts are used with interactive forms.

Web server. A Web server is a server on the Internet that holds Web documents and makes them available for viewing by remote browsers.

Server. The server is the computer in a client-server architecture that supplies files or services. The computer that requests services is called the client. The client may request file transfer, remote logins, printing, or other available services.

Browser. A browser is a client program that allows users to read hypertext documents on the Web and navigate between them. Examples are Netscape Navigator, Lynx, and Microsoft Internet Explorer. Browsers can be text-based or graphic.

Connection speed. The connection speed is the amount of time it takes to connect to a URL.

Intranet. An intranet is a local area network (LAN), which may not be connected to the Internet but which has similar functions. Some organizations set up Web servers on their own internal networks so employees have access to the organization's Web documents.

Encryption. Encryption is a procedure used in cryptography to convert plain text into ciphertext to prevent any but the intended recipient from reading that data.

Passwords. A password is a word or code used to serve as a security measure against unauthorized access to data. The computer can only verify the legitimacy of the password, not the legitimacy of the user.

Dial-up connection. A dial-up connection is a two-wire line (as used in the dial-up telephone network) that allows your computer to connect to a server via a modem.

Security. Security is the protection of data against unauthorized access. Programs and data can be secured by issuing identification numbers and passwords to authorized users of a

computer or server.

Following are additional considerations that environment engineers need to incorporate if the Web site offers special network services:

Email. Email is the transmission of memos and messages over a network. Within an enterprise, users can send mail to a single recipient or broadcast it to multiple users. With multitasking workstations, mail can be delivered and announced while the user is working in an application. The email system requires a messaging system that provides store and forward capabilities and a mail program that provides the user interface with send and receive functions.

Newsgroup. A newsgroup is a discussion group on the Internet and is focused on a particular topic. Discussions take place by posting messages for everyone to read, having online conversations, and sending email messages to individuals or the group.

Chat. Chat is a real-time conferencing capability between two or more users on a LAN, on the Internet, or via a bulletin board system (BBS). The chat is accomplished by typing on the keyboard, not speaking. Each keystroke is transmitted as it is pressed. The speed of the chat depends upon many variables.

Video conferencing. Video conferencing is a video communications session among three or more people who are geographically separated. Federal, state, and local governments are making major investments in group video-conferencing for distance learning and telemedicine.

NOTE Do you have hardware and software that support your testing? You must make sure that any software you use will be compatible with the hardware you are using and any other software packages.

The Test Bed

The *test bed* is the testing environment used for all stages of testing. Some companies may have test labs, and others may have to do station testing.

The engineers must make sure that all the software, hardware, and firmware is identical to that in which the application was developed. If your data is stored on SQL Server 7.0, the test server must also be SQL Server 7.0. All software, hardware, and firmware must be upgraded or replaced to match the environment you as the tester want recreated. Replacing and upgrading the software, hardware, and/or firmware can be costly, so make sure you have budgeted for this. The next step is for the engineer to itemize what is physically needed to create the test bed. Figure 6.3 is a template with some example data.

Name of Application:	Date:	Lead Engineer Assigned to Project:
Dates Application Will Be Tested:	Anticipated Problems:	
Dates for Setting Up Test Bed:	Engineers Assigned to Project:	Additional Resources:
Software/Hardware	Version/Type	Problems
Operating System	UNIX 4.0	
Compilers	JAVA	
Code Auditors		
Dynamic Path Drivers		
Test Drivers		
Test Data Generators	Visual Test 5.0	
MS Office	2000	
Computer	486 or above	
Monitor	Compatible with Computer	
Monitor	Color	
Interface Devices		
SQL Server	7.0	Obtain white papers
Credit Card Scanner	NCR	Need to get permission
Browser	Internet Explorer 4.0	
Browser	Netscape 4.0	

Figure 6.3 Example test application template.

Now it is time to build the test bed. The commercial software and hardware will be installed according to specifications and modifications to fit the Web site. You may want to run some preliminary installation tests that exercise the software on its platform before you begin your actual test. All commercial software and hardware should be installed in accordance with the vendor instructions or by a vendor representative. If you buy a product and plan to spend a considerable amount of time and resources on it, it is best to have a representative from that company install it and to purchase a maintenance package that includes upgrades with installations. This will also become important when you are setting up your Web site environment. The vendor is a valuable resource and can provide you with detailed specifications that will benefit your specific site. A good way to keep track of your testing environment specifications is to create a form that will document the environmental setup. Figure 6.4 illustrates an example form created in Microsoft Access. With this database you can create and track forms for your testing environment.

A Web application can be anything from an individual site that provides information and a few links to a major ecommerce site like Amazon. Think about

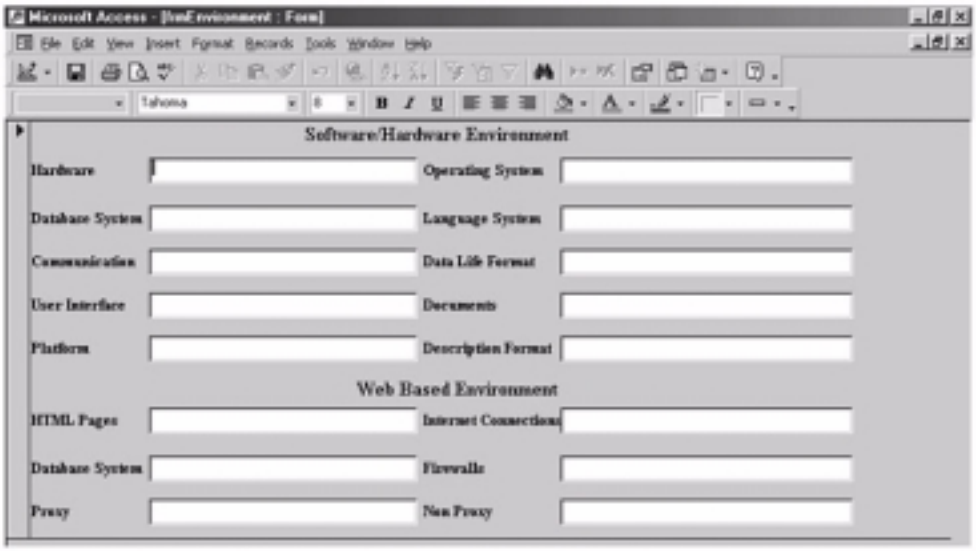

Figure 6.4 Example test environment template.

how that site actually works and how it would be tested. From just browsing, you can tell that it needs a database capable of holding millions of records. When using credit card information, you need to make sure that the site is secure when orders are placed. When a book is ordered, how it is processed and tracked are necessary components to test and also to understand in setting up your environment. If a customer orders a book and is notified by email that the book has been sent and will arrive on a selected day, how does this process affect the environment?

Amazon also includes an auction and links to other sites. If you were to recreate this system, you would want to make sure you had qualified engineers who understand the ins and outs of the site. They would need to set up the test bed so that all of the functionality could be tested. You would also want to make sure you had a qualified tester who understands the environment and how the environment will affect the testing.

To examine the entire process, let's look at a site that provides dial-up access to validate a calling card (see Figure 6.5). In the following section an example application will give you an idea of how to test dial-up capabilities.

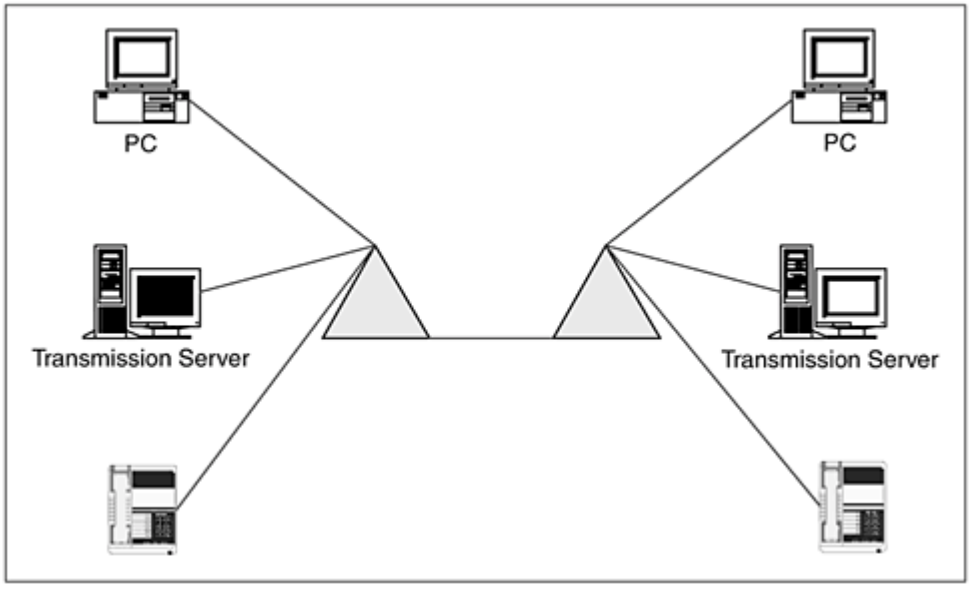

Figure 6.5 Dial-up Internet connection.

Example Application

The structure and architecture of an Internet application is essential in creating an effective Web site. For all of this to come together, you need to understand how different Web sites with different capabilities work. The example in this section shows how to set up a dial-up environment that will validate a calling card.

Because protocol message flow varies, a single diagnostic approach to AIN and INAP protocol verification cannot be used. Internap (INAP) is a leading provider of high-performance Internet connectivity services targeted at businesses seeking to maximize the performance of mission-critical Internet-based applications. It is known as Applications of Prolog. The AIN is an *Advanced Intelligent Network*, an example of which is the *public switched telephone system* (PSTN). The AIN provides enhanced voice, video, and data services and dynamic routing capabilities by using two different networks.

The diagnosis of the intelligent network (IN) at the protocol level requires a powerful protocol analysis platform that must provide manufacturers and service providers with a highly flexible tool set. The IN uses the Signaling System No. 7 (SS7) signaling protocol in which voice calls (or modem data) travel through circuit-switched voice switches, and control signals travel over an SS7 packet-switched network. The tool set will allow the creation of any AIN and INAP protocol message content and flow for simulation, capacity, validation, and monitoring purposes.

The goal is to achieve a *test creation environment* (TCE) that is analogous to the *server creation environment* (SCE). A TCE can provide the user with an environment that will support the creation, management, and execution of different AIN and INAP test cases. The TCE allows the user to create test cases to match any protocol message that will flow from the service logic programs that are executed by the *service control point* (SCP).

Table 6.1 lists verification questions that can be used to develop your scripts.

To get a clear view of the testing, let's look at a calling card in an IN environment. To test you must allow a validation to take place. The calling card service allows a user to dial a called number with a prefix such as *1. Once the number is dialed, the user must provide the service with a valid calling card

Table 6.1 Fundamental IN Protocol Verification Questions

QUESTION	DEFINITION	SOLUTION
Which message should be used?	This deals with the definition, structure, and content of the AIN or INAP messages that are used in the test case.	The TCE will allow the user to create a collection of identified messages.
How should these messages be transmitted or received?	This involves defining the sequence in which the messages are to be received or transmitted during the test execution.	The TCE handles this by providing a sequence creation service that is based on the finite machine concept.
Which set of data is to be manipulated?	When the message sequence is identified, it is considered static or dynamic.	The TCE will allow the user to assign static data values. The TCE will also allow operations that will handle the dynamic data in real time during the test execution.

number. Some services verify and validate the number and then require a personal identification number (PIN). If the calling card number and PIN are correct, the call is connected. To test this, the TCE would be used to simulate originating the *service switching point* (SSP). The first step would be to identify a set of protocol messages, such as those that follow, that would be exchanged during the transaction:

- Initial detection point
- Play announcement and collect digits
- Information collection
- Play announcement
- Proceed
- Clear call

Once these messages are designed, the sequence of the messaging must be addressed. To address these messages the following scenarios must be taken into consideration in the TCE:

- Invalid calling card number is entered.
- Valid card is entered and no PIN is required.
- Valid card is entered and a PIN is not required.
- An invalid PIN is entered.
- The call is connected.

Now that you have all your scenarios in place, you must come up with the hierarchical sequence in which each response must take place. You could use a hierarchy diagram to illustrate how you would move up and down from one step to another. It is important to understand your setup because you must build an environment to accommodate the testing and the test environment.

The TCE state machine will provide a programming environment and allow for the program to perform decision making, looping, and branching. This will enable the complete construction of the protocol message flow based on the responses of the SCP. Once this is all set up, you need to create the data that will be used for the testing.

The next item that must be checked is the database that will track the information. The environment team should make sure that the database is accessible for the hardware and software that is being used for the Web site project. A possible schema for the database is shown in Figure 6.6.

The test database can provide you with the dynamic data that will be imported for the protocol messages. By carefully selecting the test data, the tester can anticipate errors, record present errors, and understand where an error took place. It is important to understand the design of the Web site project, whether it is an IN or Internet-based. The correct environment setup will allow the tester to emulate the exact pieces that will make up your Web site.

Test Environments

Tables 6.2 through 6.5 illustrate different types of environments and provide the basic information needed to set up the software and hardware.

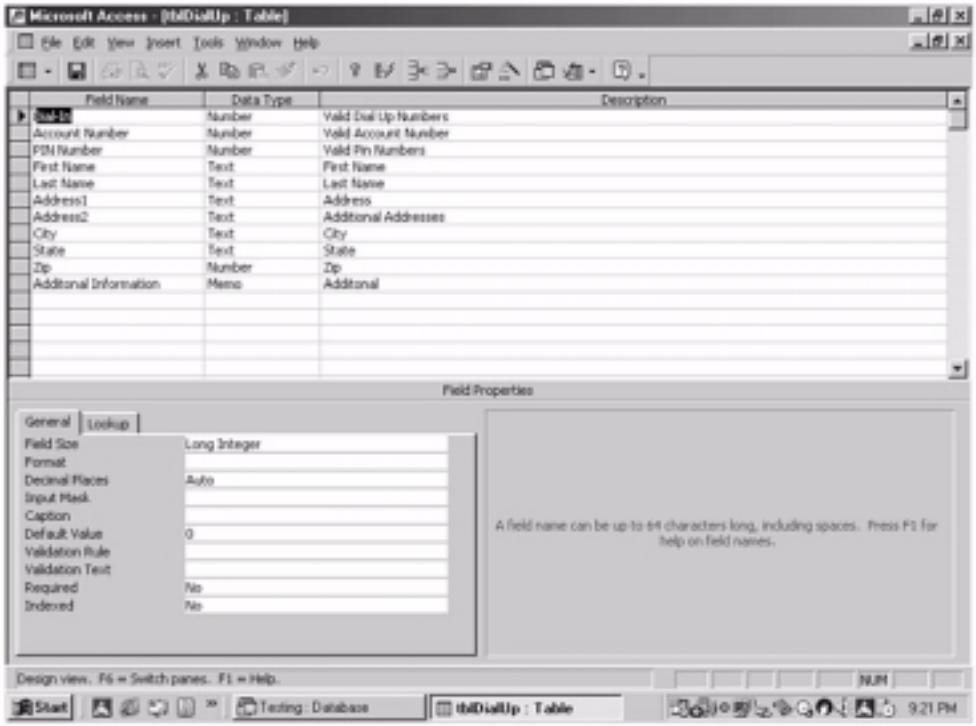

Figure 6.6 Possible database schema.

Table 6.2 Rep Server Machine

HARDWARE OR SOFTWARE	DESCRIPTION OR VERSION	DATE	SIZE IN KB
Compaq Proliant 3000	2 Pentium II 400-Mhz processors; 320 megabytes (MB) RAM	N/A	N/A
Microsoft Windows NT Server	4.00 Service Pack 5	N/A	N/A
Actuate e.Reporting Suite	4.0	N/A	N/A
Sybase Adaptive Server IQ 12 ODBC Driver	6.00.03.3005	5/3/00	426

Table 6.3 ASIQ Server Machine

HARDWARE OR SOFTWARE	DESCRIPTION OR VERSION	DATE	SIZE IN KB
Compaq Proliant 3000	2 Pentium 300-MHz processors; 320 MB RAM	N/A	N/A
Microsoft Windows NT Server	4.00 Service Pack 4	N/A	N/A
Sybase Adaptive Server IQ	12.4.2	N/A	N/A

Table 6.4 Web Server Machine

HARDWARE OR SOFTWARE	DESCRIPTION OR VERSION	DATE	SIZE IN KB
Compaq Proliant 3000	4 Pentium 200-MHz processors; 1 gigabyte (GB) RAM	N/A	N/A
Microsoft Windows NT Server	4.00 Service Pack 4	N/A	N/A
Microsoft Internet Information Server	4.0	N/A	N/A
Actuate ReportCast	4.0	N/A	N/A

Table 6.5 Client Machine

HARDWARE OR SOFTWARE	DESCRIPTION OR VERSION	DATE	SIZE IN KB
Dell Precision 410	Pentium II 450-MHz processors; 128 MB RAM	N/A	N/A
Microsoft Windows NT Workstation	4.00 Service Pack 4	N/A	N/A
Microsoft Internet Explorer	4.0	N/A	N/A
Netscape Navigator	4.7	N/A	N/A

Firewall Testing

One of the most challenging environments to test is the firewall environment. The following sections address firewall test functions in a test environment (courtesy of the Carnegie Melon Web site at www.cert.org/security-improvement/practices/p060.html).

Test the Firewall Functions in Your Test Environment

The first step is to establish a test configuration so that your firewall system is connected to two isolated hosts: the external world and your internal hosts. The default gateway for the internal host is set to the firewall system under test. If you have chosen an architecture that supports centralized logging, place both the internal host and a log host on your internal network so that you can test logging options. If logging is performed on the firewall host, you can connect the internal host directly to the firewall host. If you are using a Unix-based system, you may want to control the logging in and logging out with a Unix shell script that will authorize and authenticate the users.

After scanning or network sniffing tools are in place on your outside and inside hosts to capture all traffic in both directions (inside to outside, outside to inside), you can then perform the following steps:

1. Disable packet filtering.
2. Inject packets that will exercise all routing rules and send these through the firewall system.
3. Ensure packets that are routed correctly by examining the firewall logs and your scanner results.
4. Turn on packet filtering.
5. Inject network traffic that is an appropriate sampling of all possible source and destination IP addresses, across all ports, and for all protocols.

1. Ensure packets that are intended to be blocked (denied) are blocked. For example, if all User Datagram Protocol (UDP) packets are to be blocked, ensure that none get through. Ensure that packets permitted to enter or exit do enter and exit. Do this by examining your firewall logs and scanner results.
2. Scan for open and blocked ports to ensure your firewall system is performing as intended.
3. Examine all of the network traffic that is logged and verify that the logging options associated with each packet-filtering rule are operating as intended.
4. Examine all of the network traffic that is logged and verify that the alert options associated with each logging option are sending alerts to the designated destination (such as the firewall administrator) using the specified mechanism (such as paging or email).

It would be a good idea to plan to conduct this step and the next step with at least two people. The first should be the implementer of the routing configuration, packet-filtering rules, logging options, and alert options. The second should be someone who reviews what has been implemented, understands the intent, and agrees that the network topology and security policies have been reflected correctly.

Test the Firewall Functions in Your Production Environment

The next step assumes that you are migrating from single-layer firewall architecture to a multiple layer. It also assumes that you have a network topology of one or more private networks and one or more public networks. The public networks typically connect hosts that respond to internal and external requests for service, such as WWW (HTTP), FTP, email (SMTP), and DNS. These hosts may also respond to internal requests for service such as SNMP, file access, and logging. The public network as described here can serve as your Demilitarized Zone (DMZ). Demilitarized Zones are used by companies that want to host their own Internet services without sacrificing unauthorized access to their private network. The DMZ sits between the Internet and an internal network's line of defense, usually some combination of firewalls and bastion hosts. The

private network typically connects hosts that service your internal users including individual user workstations. You will then be ready to perform the following steps:

1. Connect your firewall system to your public and private networks.
2. Set the routing configuration on selected public and private network hosts to direct traffic through the firewall system. The basis for selection is on a service-by-service

basis, for example, the Web server on your public network and the host storing the files that the Web server needs to access on your private network. Cycle through the selection and exercise of all services such as Web, file access, DNS, mail, and logging. Log the firewall system's incoming and outgoing network traffic. Use a scanner or network sniffer to observe what is happening.

3. Ensure packets that are intended to be blocked (denied) are blocked. For example, if all UDP packets are to be blocked, ensure that none get through. Ensure that packets permitted to enter or exit do so. Do this by examining your firewall logs and scanner results.

4. Scan all hosts in a selected portion of your network that includes the firewall system. Verify that you cannot gain any undesired information due to the scanning packets being blocked (denied). Attempt source port scanning using a well-known port such as the FTP data port (port 20) to ensure that you cannot use the port for a service other than the one intended.

5. You can use intrusion detection system tools in a simulated or live network traffic test to aid you in determining if your packet-filtering rules are protecting your systems and networks from known attacks. You will need to run these tools for some period of time and review the results on a regular basis. You may want to defer this level of testing to normal operations once you have fully deployed the new firewall system.

6. Examine all of the network traffic that is logged and verify that the logging options associated with each packet-filtering rule are operating as intended.

7. Examine all of the network traffic that is logged and verify that the alert options associated with each logging option are sending alerts to the designated destination (such as the firewall administrator) using the specified mechanism (such as paging or email).

You cannot do a final test of your routing configuration prior to connecting the firewall system to your operational external interfaces. As a result, you should run live packets through your internal networks using the new firewall system prior to connecting to the outside world (to the greatest extent possible). To mitigate the risk of unexpected problems in this final test phase, you should initiate the operational connections for a small subset of hosts (such as those used by your system and firewall administrators) prior to connecting large numbers of user workstation or server hosts.

Summary

Because Web sites are more than just viewing an URL in a browser, your test team must be sharp and understand today's technology. Web sites can have dial-up features, audio features, video conferencing, blackboards, and chats. Each test environment should be set up according to the needs and requirements of the software and hardware you will be using to run your Web site.

Chapter 7 will discuss testing languages and databases.

CHAPTER 7
Testing Languages and Databases

Testing Web applications can be a challenge. There are many different components that make up the Web application, such as the environment, network, database, language, and browser interface. Web technologies such as HTML, Java, JavaScript, and VBScript are just a few of the Web tester's concerns. It is a good idea for the tester to understand a variety of languages, networks, and databases.

We have looked at the Web testing process and Web testing tools. In this chapter we will look at the different types of languages and the testing process that will assist you with all the different types of Web components you may come across.

Java

Java is one of the most successful languages for Internet programming to date. It has distinct characteristics that set it apart from other programming languages. It is distributed, interpreted, and robust. Java is also a portable

language that is capable of running on many different types of machines. Java was designed to adapt to the continually evolving environment of the Internet. This language can dynamically load in classes when they are needed. It is these classes (as well as threading) with which a tester is concerned. With Java's object-oriented components such as inheritance, encapsulation, and polymorphism, it is important to understand these concepts before designing a test.

When you are approaching the test from a module view, you need to test with a top-down approach by simulating each module and working toward an integrated approach. This type of testing will require the tester to write stubs to simulate modules that may affect modules lower in the program. The stub is nothing more than a simulated version of the working modules that are necessary to test the particular module with which you are working. These stubs can simulate the activity, allowing you to locate potential errors before all modules are integrated.

There are many different types of Java testing tools available that will test classes, applications, and applets. A tool worth considering is Jtest, which can call a class (or classes), check for compile class errors, and build a set of tests. Testing at the class level is critical because that is where full coverage of the methods is easiest to achieve. When you test a class apart from all other objects, it becomes significantly easier to correct potential errors because you are much closer to the errors. Figures 7.1 and 7.2 illustrate some of the features of Jtest.

Scripting Languages

As you create a Web page, you may feel that scripting languages like VBScript or Java Script are needed to address some of your needs. A scripting language can develop client and server Internet applications. For example, Netscape Navigator 2.0 interprets JavaScript statements

that are embedded directly in an HTML page. In a client application for Navigator, JavaScript statements embedded in an HTML page can recognize and respond to user events such as mouse clicks, form input, and page navigation. There are also other Web scripting languages available, such as Perl and CGI. Microsoft has two scripting languages, VBScript and JScript, which ship as an ActiveX scripting language engine in Internet Explorer and the Internet Information Server (IIS). Netscape has JavaScript, a cross-platform language that is associated with dynamic Web

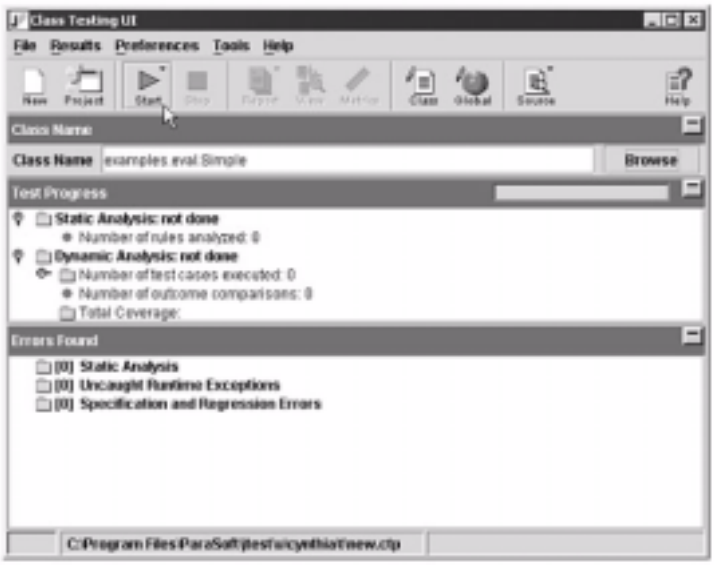

Figure 7.1 Example of Jtest.

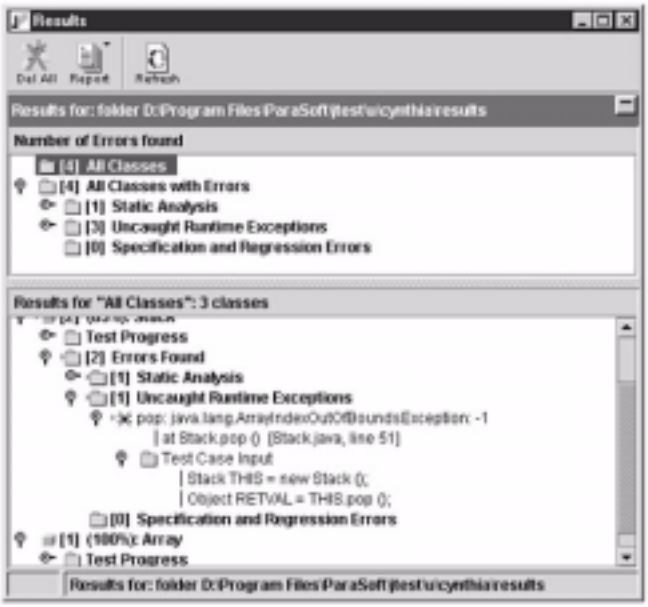

Figure 7.2 Another example of Jtest.

pages. The following sections will discuss these languages, how they are used, and their different syntaxes.

VBScript

VBScript is a member of the Visual Basic family. VBScript brings active scripting to a wide variety of environments, including Web client scripting in Microsoft Internet Explorer and Web server scripting in Microsoft Internet Information Service. If you already know Visual Basic or Visual Basic for Applications, the VBScript syntax will look very familiar. Even if you do not know Visual Basic, once you learn VBScript, you are on your way to programming with the Visual Basic languages.

Following is an HTML example with two features. These features tell the browser through HTML tags what the input is and then what the value of the input is.

```
&INPUT TYPE=BUTTON VALUE=∀Click me∀ NAME=∀BtnHello∀>
```

You can use a &SCRIPT> tag that contains an event handler for the BtnHello OnClick event. Event handlers are named using the pattern ObjectName_EventName, in this case, BtnHello_OnClick. When the button is clicked, the procedure with that name is run. In response to the button being clicked, the event handler displays a message box by using the VBScript MsgBox run-time function. Following is an example of VBScript:

```
&SCRIPT LANGUAGE=∀VBScript∀>
    Sub BtnHello_OnClick
        MsgBox ∀Hello World!∀, 0, ∀My first active document∀
    End Sub
&/SCRIPT>
```

To test the buttons you could use code similar to the following:

```
&FORM NAME=∀InputForm∀>
&INPUT TYPE=RADIO NAME=∀ImageSet∀>Image Set 1
&INPUT TYPE=RADIO NAME=∀ImageSet∀>Image Set 2
&INPUT TYPE=RADIO NAME=∀ImageSet∀>Image Set 3
&INPUT TYPE=RADIO NAME=∀ImageSet∀>Image Set 4
&/FORM>
```


Then you can initialize any or all buttons with:

```
InputForm.ImageSet.Item(n).Checked = TRUE (where n starts from 0)
```

and test values with:

```
If InputForm.ImageSet.Item(0).Checked Then . . .
```

You can use this method of scripting to test the different types of scripts.

Following is information to keep in mind as you test VBScript:

Error handling. VBScript includes the On Error Resume Next statement for handling exceptions. If you are writing server script, error handling is particularly important

because the script runs unattended. This allows the testers to check for errors and track errors as they occur.

Formatting. VBScript includes functions that make it easy to format date, number, and currency data. Being able to format certain data ensures that the user only enters data in a certain manner; this also allows testers to track how data is entered and verify the integrity of the data.

Easier event handling in Internet Explorer. VBScript allows you to create an "implicit" event handler by simply naming a function with the syntax *object_event*—for example, Button1_onclick. VB Script can only be run through a known browser, so testers must be aware of the environment in which the scripts will be run.

JScript

Microsoft JScript 5.5 is the first scripting language to fully conform to ECMAScript, the Web's only standard scripting language. JScript is the Microsoft implementation of the ECMA 262 language specification (ECMAScript Edition 3). With only a few minor exceptions (to maintain backward compatibility), JScript is a full implementation of the ECMA standard. The ECMAScript standard describes a Web scripting language that can enrich and enliven Web pages in a Web browser. ECMAScript is the only standard scripting language on the Web; it is based on the ECMA-262 specification,

which outlines an object-oriented programming language for performing computations and manipulating objects within a host environment, such as the browser. The complete ECMA-262 specification can be found at www.ecma.ch/stand/ecma-262.htm.

JScript is an interpreted, object-based scripting language. Although it has fewer capabilities than full-fledged object-oriented languages like C + +, JScript is more than sufficiently powerful for its intended purposes. It is not a cut-down or simplified version of another language (it is only distantly and indirectly related to Java, for example). You cannot write stand-alone applications in it because it has no built-in support for reading or writing files. JScript scripts can run only in the presence of an interpreter or *host,* such as Active Server Pages (ASP), Internet Explorer, or Windows Script Host. You cannot explicitly declare data types in JScript. For instance, if you add a number to an item consisting of text (a string), the number is converted to text.

The code in the following example is more explicit and less dense than code you are likely to find in actual Web pages. The intent is to clarify the concepts, not to express optimal coding conciseness and style. In any case, there is no shame in writing code that you can read and easily understand six months after you write it. This example is for an HTML page that has two salient features. The &INPUT> tag creates a button named BtnHello and OnClick identifies the code that should be run when the button is clicked.

```
&INPUT TYPE=BUTTON VALUE=∀Click me∀ NAME=∀BtnHello∀
OnClick=∀sayhello()∀>
```

The &SCRIPT LANGUAGE> section contains Microsoft JScript code, which defines the

Sayhello function. To run the Sayhello function, click BtnHello, which causes an Alert dialog box to be displayed. The browser supplies the Alert method.

```
&SCRIPT LANGUAGE=∀JavaScript∀>
 &!--
 function sayhello ()
 {
    alert(∀Hello world!∀)
 }
 //-->
&/SCRIPT>
```


NOTE JScript is case sensitive, which may cause errors.

Following are some JScript issues to keep in mind:

Check that your server supports JScript. IIS allows you to use JScript for scripting ASP pages. The tester needs to be aware of the browser to make sure that it supports JScript.

Dynamic execution. A very powerful feature of JScript is that it allows you to create and execute script or evaluate expressions dynamically from within your script. Your script can write script. This feature is handy when working with dynamic HTML (DHTML) because it allows you to dynamically manipulate the DHTML document model, see what the script is doing, and then evaluate the DHTML document model.

Object orientation. JScript uses a prototype-based object structure that allows you to define objects in a script. You can extend both built-in and custom-built objects by adding methods and properties to the objects' prototypes. The tester must be aware of how the objects are run and their properties so if the objects are not responding to the unit test, the tester can report the error to the developer.

JavaScript

JavaScript is Netscape's cross-platform, object-based scripting language for client and server applications. With JavaScript you can create applications that run over the Internet. Client applications run in a browser, such as Netscape Navigator, and server applications run on a server, such as Netscape Enterprise Server. Using JavaScript, you can create dynamic HTML pages that process user input and maintain persistent data using special objects, files, and relational databases. Through JavaScript's LiveConnect functionality, your applications can access Java and CORBA distributed-object applications.

Server- and client-side JavaScript share the same core language, which corresponds to ECMA-262, the scripting language standardized by the European standards body, with some additions. The core language contains a set of core objects, such as the Array and Date objects. It also defines other language features such as its expressions, statements, and operators. Although server- and client-side JavaScript use the same core functionality, in some cases they use them differently.

The following script opens up a record set from a data connection. The script tells the ASP to create and show the record. Once it creates the record, you can use HTML to view the record that the script called up.

```
&!--#INCLUDE FILE=∀_ScriptLibrary/Recordset.ASP∀-->
&SCRIPT LANGUAGE=∀JavaScript∀ RUNAT=∀server∀>
function _initRecordset1()
{
    thisPage.createDE();
    var rsTmp = DE.Recordsets('Command2');
    Recordset1.setRecordSource(rsTmp);
    Recordset1.open();
    if (thisPage.getState('pb_Recordset1') != null)
        Recordset1.setBookmark(thisPage.getState('pb_Recordset1'));
}
function _Recordset1_ctor()
{
    CreateRecordset('Recordset1', _initRecordset1, null);
}
function _Recordset1_dtor()
{
    Recordset1._preserveState();
    thisPage.setState('pb_Recordset1', Recordset1.getBookmark());
}
&/SCRIPT>
&!--METADATA TYPE=∀DesignerControl∀ endspan-->
```

Testing Scripting Languages

When testing scripting languages, there are several tools available and several issues that need to be addressed. Microsoft's Script Debugger can be used to debug and edit scripts. This tool can be used for a step through when doing a unit test or can be used by the developers to debug scripts as they are written.

The Microsoft Script Debugger is a debugging environment that extends any Microsoft ActiveX scripting host. When used in conjunction with IE and IIS, Web developers can browse, edit, and debug scripted HTML pages (.HTM, .HTML, and .ASP files) on both the client and the server. It allows developers to more efficiently and effectively develop script applications and provides these features:

- View the source code of the script being debugged
- Control the pace of script execution with break points and stepping
- View and change variable and property values with the Command window

- View and control script flow with the Call Stack window

Developers can save time and build powerful Active Server Pages if they can debug scripts as they are written or test the scripts after they are published in a unit test.

Testers may want to do test script pages to test the script. Once the page is created, they can test it in a browser to see if the script is accepted and the page looks the way it should.

Following are the steps and the possible code you may want to use:

1. Create a space in memory called Message.
2. Set the value of Message to "Hello Tester"
3. Create a dialog box containing the value of Message.
4. End the routine.
5. End the comment.
6. End the script.

Once you have your plan for your page, you can create a Web page with the following script:

```
&HTML>
&HEAD>
&TITLE>Tester Page&/TITLE>
&/HEAD>
&BODY>
&H1>Tester Page for VBS&/H1>
&HR COLOR=∀RED∀>
&INPUT TYPE=∀SUBMIT∀ NAME=∀Btn1∀ VALUE=∀Click here to test the code∀>
```



```
&SCRIPT LANGUAGE=∀VBS∀>
&!--
Sub Btn1_OnClick()
Dim Message
Message=∀Hello Tester!∀
MsgBox Message, 0, ∀Tester Result∀
End Sub
-->
&/SCRIPT>
&/BODY>&/HTML>
```

The browser will read the file just as it would with any HTML document. The button created on the page is no different from the buttons that are created for forms. The &H1> text shows

in the browser, followed by a horizontal rule, and finally the button. The browser now hits the &SCRIPT="VBS"> tag. Here the browser hands off to the scripting engine, which in turn hands off to the VBScript (VBS) interpreter. The VBS code is then parsed by VBS, compiled on the fly, and run. The results will then appear in the form of a message box. If the test works, the tester knows the browser will accept the scripting language in which the page was written.

Databases

Performing Web database testing is similar to performing database testing except that you need to account for Internet server issues. Keep in mind that a database is a place where data is stored in an organized manner. You can use a database to track what you are testing, or you can use it as a part of your Web application.

The database chosen for use in testing will depend on the data access requirements of your Web application. The Jet engine (Joint engine technology, the database engine used by Microsoft Office and Visual Basic) is behind what is sometimes referred to as a personal database; it can handle a moderate number of simultaneous users. If your application is small and does not require the benefits of a large-scale database, you might want to test (and ultimately implement) your application using Jet. Microsoft Access 97 is considered a Jet database.

If you are modeling an enterprise application that requires high functioning and will store a lot of data, you should test with SQL Server 7.0. Using it, you can actually modify the tables and queries to represent the application's final data access requirements. SQL Server 7.0 can be used to house the application's data.

In SQL Server 7.0 you can write stored procedures that will generate SQL statements that let you view your data in the manner that is useful for your business. You can use Web publishing directly from SQL Server 7.0, or you can use a tool like InterDev or FrontPage 2000 to connect your database. If you are going to use a connection string, make sure that ActiveX Data Objects (ADO), Object Linking and Embedding Databases (OLE DB), and Open Database Connectivity (ODBC) are understood so that you use what is best for your application.

The following sections detail some important issues that apply to using your database as part of your Web application.

Database Testing

In a business application, an ecommerce site typically stores catalogs, shopping baskets, user profiles, and order information in the database. Testing a database requires the ability to check and verify the data that is housed in the database; the SQL language can be used to query tables, columns, and rows.

Your database can be located on the server on which it is stored or on a server that is protected by a firewall. If a firewall is used, it will add complexity to the testing processes because you will have to perform accessibility, security, and performance testing.

Objective of Database Testing

The objective in testing your database is to determine how well the database meets the business requirements. This process is ongoing because databases are not static. When a database is created, a parallel, or mirror, database should be created. Store the parallel database on another computer in case one computer encounters performance problems. The original database is left alone and the parallel one goes through the various tests. This process continues until the tests are successful; then changes can be implemented in the original database.

To accommodate the special functionality of a database-driven Web site, it is essential to focus on load, stress, usability, link, and validation testing. Databases are tested for five different reasons:

- Relevance of search results
- Query response time

- Data integrity
- Data validity
- Recovery

Table 7.1 illustrates some items that the tester must be aware of when doing Web database testing.

Relevance of Database Search Results

The search option is one of the frequently used functions of online databases. Search results provide direct links to other pages, saving time and effort. They also give the user immediate results. Many site visitors complain that search results are not relevant to the Web site. Therefore, it is important to build relevance into database searches. This feature should be tested before production and should become part of the ongoing test process. It should also be part of the data handling.

A team of users who are not a part of the development team should test for search relevance. The team will act as online customers and should try out random search options using different keywords. The search results should be recorded by relevance to the keyword. As testing progresses, metrics can be applied to measure results. A percentage can be applied for the correct results achieved versus the expected results. At the conclusion of the testing, the team should come up with a series of recommendations for the developers that can be incorporated into the database search options.

Table 7.1 Items to Check When Testing a Database

WHAT TO TEST	ENVIRONMENT	TOOLS/TECHNIQUE
Search results	System test environment	Black Box and White Box technique
Response time	System test environment	Syntax Testing/Functional Testing
Data integrity	Development environment	White Box Testing
Data validity	Development environment	White Box Testing

Query Response Time

The turnaround time for responding to queries in a database must be short; therefore, query response time is essential for online transactions. The results from this test will help to identify problems, such as possible bottlenecks in the network, specific queries, the database structure, or the hardware.

Data Integrity

Data stored in the database should include such items as the catalog, pricing, shipping tables, tax tables, order database, and customer information. Testing must verify the integrity of the stored data. Testing should be done on a regular basis because data changes over time.

Data Integrity Tests

Data integrity can be tested as follows to ensure that the data is valid and not corrupt:

- Test the creation, modification, and deletion of data in tables as specified in the business requirement.
- Test to make sure that sets of radio buttons represent a fixed set of values. You should also check for NULL or EMPTY values.
- Test to make sure that data is saved to the database and that each value gets saved fully. You should watch for the truncation of strings and that numeric values are not rounded off.
- Test to make sure that default values are stored and saved.
- Test the compatibility with old data. You should ensure that all updates do not affect the data you have on file in your database.

Data Validity

The most common data errors are due to incorrect data entry, called data validity errors. These errors are the hardest to detect in the database system. They are typically caused when a large volume of data is entered in a short time frame. For example, $78 can be entered as $87

by mistake. The data entered is invalid and cannot be used.

Data validity errors can be prevented if you use the data validation rules in the data fields. For example, the date field in a database uses the MM/DD/YYYY format. A developer can incorporate a data validation rule, such as MM does not exceed 12, and DD does not exceed 31. In many cases, simple field validation rules are unable to detect data validity errors, in which case queries can be used to validate data fields. For example, a query can be written to compare the sum of the numbers in the database data field with the original sum of numbers from the source. A difference between the figures indicates an error in at least one data element.

Recovery Testing

Another test that is performed on database software is the recovery test. This test involves forcing the system to fail in a variety of ways to ensure that:

- The system recovers from faults and resumes processing within a predefined period of time.
- The system is fault-tolerant, which means that processing faults do not halt the overall functioning of the system.
- Data recovery and restart are correct in case of automatic recovery. If recovery requires human intervention, the mean time to repair the database is within predefined acceptable limits.

Example Database Environments

The following sections detail a few database environment examples, including explanations of the strong and weak points.

SQL Server

Figure 7.3 illustrates what the SQL Server 7.0 database environment looks like. Microsoft SQL Server version 7.0 is a database designed around the Windows framework. Customer needs and requirements have driven significant product innovations in ease of use, reliability, scalability, and data warehousing. SQL Server 7.0 runs on Windows NT 4.0 or Windows 2000. SQL Server 7.0 Enterprise Edition builds on the established strengths and broad functionality of the SQL Server, extending its already extensive scalability, interoperability, availability, and manageability. Enterprise Edition provides the means for build-

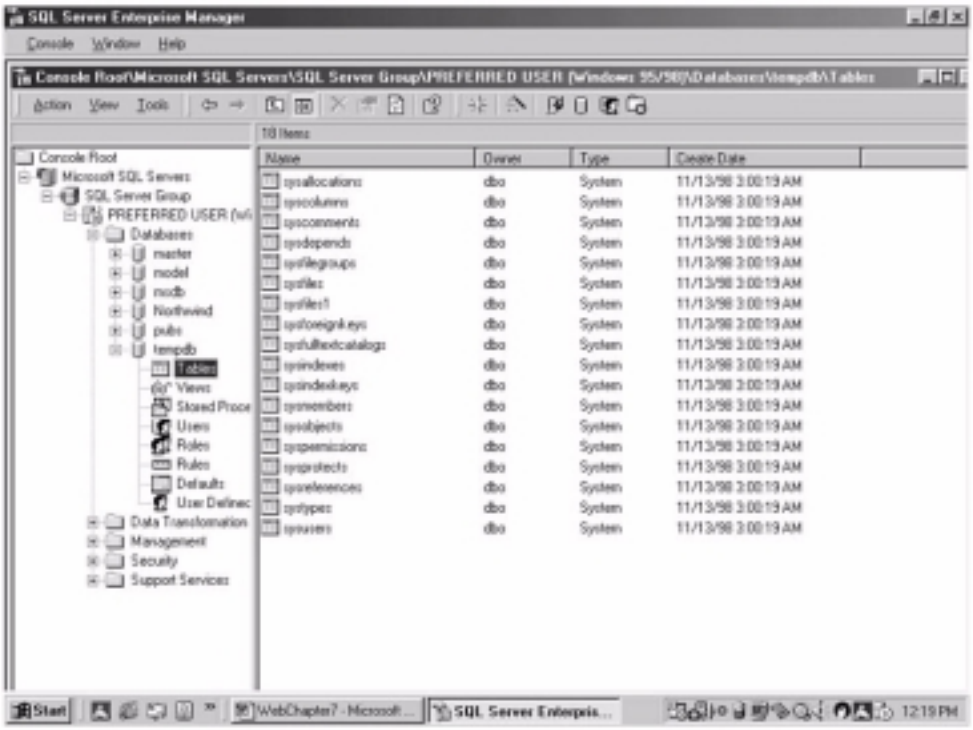

Figure 7.3 Example of a SQL Server 7.0 environment.

ing and deploying large-scale distributed applications, making it the best platform for the largest and most mission-critical database applications. The SQL Server Enterprise Edition provides clustering support and can expand to use up to 3 GB of memory; it runs on Windows NT 4.0 Enterprise Edition or Windows 2000-Advanced Server.

There are several issues that must be addressed when testing a SQL Server database. Consider the following:

- If the Web site publishes from inside the SQL Server straight to a Web page, is the data accurate and of the correct data type?
- If the SQL Server reads from a stored procedure to produce a Web page or if the stored procedure is changed, does the data on the page change?
- If you are using FrontPage or InterDev, is the data connection to your pages secure?

- Does the database have scheduled maintenance with a log so testers can see changes or errors?
- Can the tester check to see how back ups are being handled?
- Is the database secure?

Access

Figure 7.4 illustrates an example MS Access database. Microsoft Access 2000 makes it easy to get the information you need. It gives the programmer powerful tools that help organize your information and allows you to find answers that count, share information over intranets, and build faster and more effective business solutions.

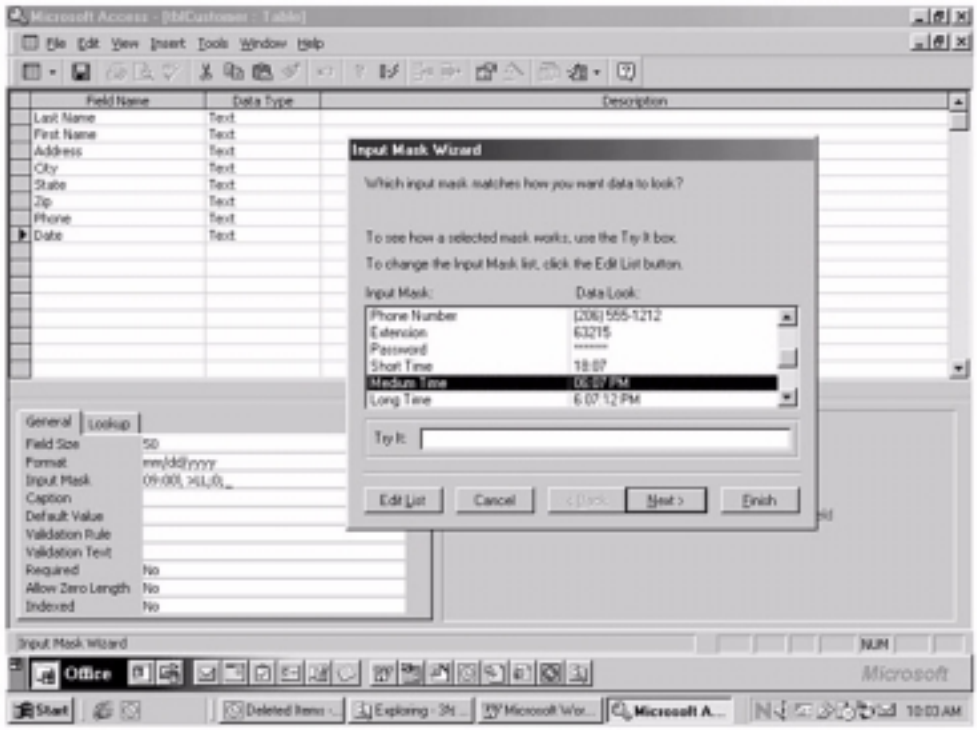

Figure 7.4 Example MS Access database.

Page 157

You can also use MS Access to enable Web collaboration. By using its Web-enhanced tools you can make data immediately available to coworkers. It allows you to use the application from your desktop or on the road. You can also customize your views and formats to show the information you need. MS Access 2000 has a built-in Microsoft SQL Server integration application to create a scalable database that can grow with your business.

Testing an Access database can be tricky. There are many tools included with Access that ensure data integrity, which is critical for accurate data. Data integrity is best achieved by applying basic normalization rules. A developer can also create and test modules and macros to ensure the integrity of the data. The following are items that a tester needs to be aware of when testing an Access database:

- If the database is creating Web pages from the database to a URL, is the information correct and updated? If the pages are not dynamic or Active Server Pages, they will not update automatically.
- If the tables in the database are linked to another database, make sure that all the links are active and giving relevant information.
- Are the fields such as zip code, phone numbers, dates, currency, and social security

number formatted properly?

- If there are formulas in the database, do they work? How will they take care of updates if numbers change (for example, updating taxes)?
- Do the forms populate the correct tables?
- Is the database secure?

FoxPro

Figure 7.5 shows a view of Visual FoxPro 6.0 that is available with the Visual Studio 6.0 package. Microsoft Visual FoxPro 6.0 is the newest version of Microsoft's tool for creating high-performance, state-of-the-art database components and solutions. Visual FoxPro 6.0 gives developers and programmers the necessary tools to manage data. It can be organized in tables and through queries and will create an integrated relational database management system (DBMS).

Visual FoxPro 6.0 is a tool that can be used for building components that can be deployed and scaled in client-server, Internet, and intranet environments.

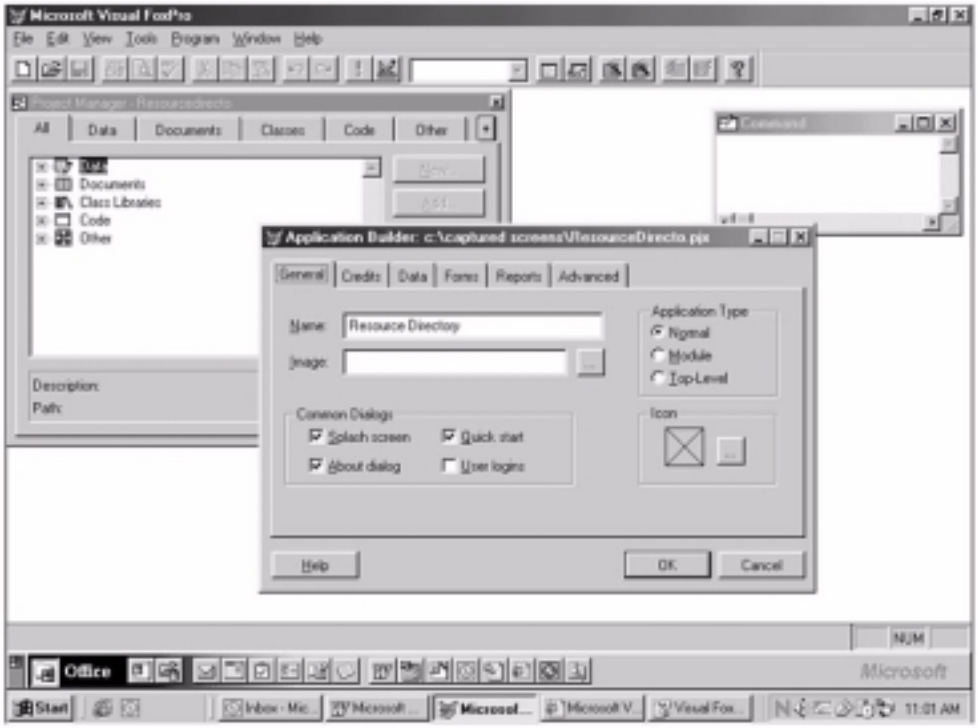

Figure 7.5 Visual FoxPro 6.0.

These components will most likely be middle-tier business rules, servers, and other components that work with local or remote data. It is easy to use and deploy. Following are some items that a tester must be aware of when testing a Visual FoxPro database:

- If the database is linked to other databases, are the links secure and working?

- If the database publishes to the Internet, is the data correct?
- When data is deployed, is it still accurate?
- Do the queries give accurate information to the reports?
- If the database performs calculations, are the calculations accurate?

Oracle

The Oracle9*i* Application Server offers an innovative and comprehensive set of middle-tier services. From self-service enterprise portals and high-traffic estores to supplier exchanges, the Oracle9*i* Application Server is the best application server for your database-driven Web sites. It contains hot new caching technology that can dramatically increase Web site performance, scalability, and availability. With this caching technology, you can support more users with richer, more personalized dynamic Web content, all without adding more application or database servers, which significantly reduces the cost of running your Web site.

Oracle9*i* Application Server delivers scalability and performance to deployments of all your Web applications. For new Web site developers, Oracle Portal services make it easy to deploy enterprise portals with central management and unified security. Or, you can deploy Web sites built in standard Java, with rich eXtensible Markup Language (XML) and content management support. You can also deploy your back-office transactional applications that were built using Oracle Forms Developer.

To extend the reach of your Web portal, Oracle9*i* Application Server enables you to make information in any database or Internet application available to any wireless device. You no longer need to worry about supporting each device's specific markup language.

Database-driven Web sites are taking an increasingly critical role in the day-to-day operations of many organizations. You can minimize your risk of site problems, errors, and failures through careful analysis of requirements and risks, well-planned site test strategies, and appropriate use of new Web test tools.

Database-Driven Web Sites

There are special considerations to think about when testing a database-driven Web site. There are new test tools that will help test the Web sites, but first it is important to understand the testing strategy. Web site testing is similar to testing standard client-server applications, but there are unique considerations that can direct the focus of your testing strategy.

The client side of a Web-based application is contingent upon any one of dozens of Web browsers installed on various operating systems and hardware platforms. Every browser will conform in varying degrees to multiple

standards, with numerous possible plug-ins, user-controlled options, and settings. It is important to remember that loads are affected with such factors as time of day, publicity, type

of connection, and site changes. System testing is difficult because there are so many variables. Refer to the section *System Testing* in Chapter 2, "Test Methodology," for more information on system testing.

Applets

Unlike applications, which are launched from the command line, applets operate within a browser context. In this environment, it is up to the programmer to fully specify the details such as font and color of any text to be displayed. Applets do not begin execution from a single main() method. Rather, they are responsive to five service functions (or methods) needed by the browser environment:

init( ). The applet initializes itself for use when it is first loaded into memory.

start( ). Actions the applet performs after initialization or whenever the browser returns to the applet.

stop( ). Actions the applet performs whenever the browser leaves the applet.

destroy( ). Operations the applet carries out to clean up after itself just before it is freed from memory.

paint(Graphics g). The mechanism by which the applet displays something within the window allocated to it by the browser.

Testing an applet begins by testing it on the desktop, and it should be tested on various browsers.

ActiveX Data Objects

ActiveX objects are an important part of today's Web. We will first explain what an ActiveX object is and then we will talk about testing ActiveX. If your developers and testers work together in the test process, testing can be done with tools provided with Visual Studio or tools listed in Chapter 5, "Web Site Testing Tools."

ADO is Microsoft's strategic, high-level interface to data that helps the programmer use the underlying OLE DB, and ODBC technologies. The ADO

supplies an open, application-level data-access object model that allows programmers to write database applications over OLE DB data using any language. By using ADO, developers have access to more types of data than ever before and will need to spend far less time writing complex client-server code. Figure 7.6 is a diagram from Microsoft that illustrates the ADO architecture.

ActiveX has a set of technologies from Microsoft that enables programmers to create interactive content for the World Wide Web. With ActiveX, Web sites come alive using multimedia effects, interactive objects, and sophisticated applications that create a user experience comparable to that of high-quality CD-ROM titles. ActiveX provides the glue that ties together a wide assortment of technology building blocks to enable *active* Web sites. An

active Web site provides the following:

- Active Web content that will attract and retain users.
- Open, cross-platform support on Macintosh, Windows, and Unix operating systems.

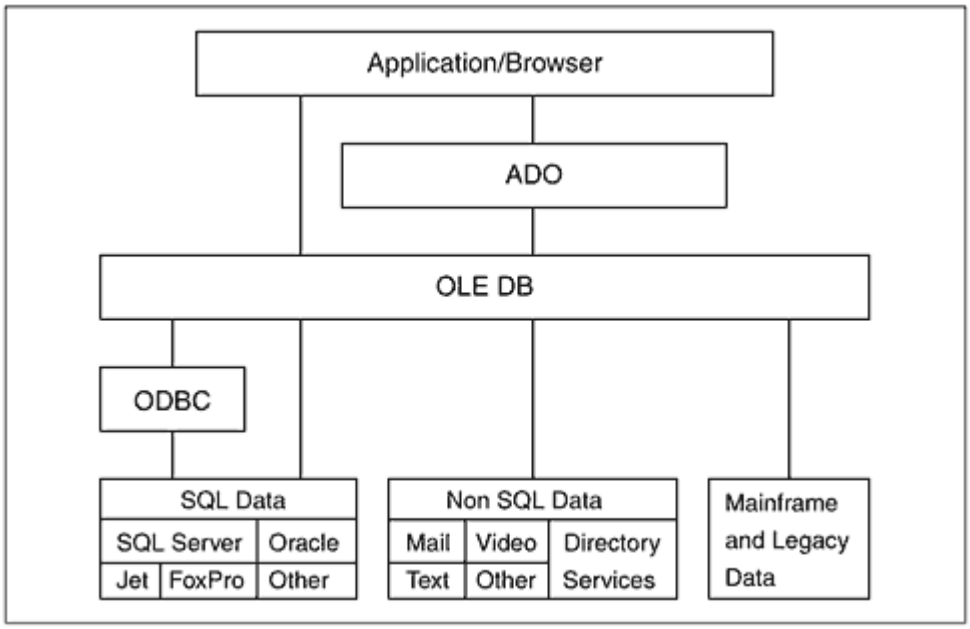

Figure 7.6 ADO architecture.

Page 162

- Familiar tools from a wide assortment of vendors, including Visual Basic, Visual C + +, Borland Delphi, Borland C + +, Java, and Java-enabled tools. Developers can use what they know and be productive immediately.
- Existing inventory of ActiveX controls available today for immediate use by Web producers.
- Industry standards, with built-in support for key industry and de facto marketplace standards, including HTML, TCP/IP, Java, COM, and others.

Other features of an active site are:

- ActiveX includes both client and server technologies.
- ActiveX controls are the interactive objects in a Web page that provide interactive and user-controllable functions and hence enliven a Web site.
- ActiveX documents enable users to view non-HTML documents, such as Microsoft Excel or Word files, through a Web browser.

- Active scripting controls the integrated behavior of several ActiveX controls and/or Java applets from the browser or server.
- Java Virtual Machine is the code that enables any ActiveX-supported browser such as Internet Explorer 3.0 to run Java applets and to integrate Java applets with ActiveX controls.
- ActiveX Server Framework provides a number of Web server-based functions such as security, database access, and others.

ActiveX brings innovation and interactivity to the Web. Because many different languages and tools support it, it enables developers with varied backgrounds and expertise to bring their creativity to the Web. ActiveX takes the most creative and innovative software development efforts and enables them to work together seamlessly in a Web site. With thousands of these software components already existing, an exciting collection of interactive objects is available for immediate use by Web producers.

ActiveX makes it fast and easy for developers and Web producers to create unique, interactive Web sites that will make the Internet fundamentally more useful and productive. Web producers don't have to start from scratch and build all the parts of their interactive Web site by hand because there are already more than 1,000 reusable controls available in the market. And because ActiveX can be used with a wide variety of programming languages from dozens of vendors, developers and Webmasters can make use of their current expertise to more quickly create compelling content. They can also accommodate a wide range of users because ActiveX is supported on multiple operating system platforms.

ActiveX provides a standard mechanism to extend any programming language, including Java. ActiveX extends the capabilities of the Java language by allowing Java developers to integrate their applets with the richness of ActiveX. ActiveX ties Java applets together with objects created in other languages, so Java programmers can link to ActiveX controls directly from their Java programs. By the same token, objects written in other programming languages from multiple vendors can link to Java applets. ActiveX ties them all together, delivering the most powerful Web technologies in an open, integrated platform. By providing a common way to extend and link programming languages, ActiveX maximizes developers' resources for interactive Web development.

Small, medium, and large software companies currently create ActiveX controls, including companies such as Borland, Oracle, and Sybase/Powersoft. As a result of their work, there are more than 1,000 existing ActiveX controls available for use today by Web producers. In addition, 14 companies that create Web design and development tools have built ActiveX support into their products, allowing their customers to both create and make use of ActiveX controls in their programs. Microsoft's Internet Explorer supports ActiveX, and Microsoft provides the ActiveX plug-in for Netscape Navigator, enabling the broadest range of Internet users to view ActiveX-enabled Web pages. ActiveX is currently supported on the Windows operating system.

Testing ActiveX Controls

One of the items that a tester must first address is the Web security problem that surrounds ActiveX controls. Security is extremely important when you're using ActiveX for Web deployment. Although ActiveX controls can be very powerful

and convenient, they can also become your worst nightmare. As a builder of ActiveX controls, consider the following:

- Don't build ActiveX controls that are harmful.
- Build your controls to be tamper-resistant.
- Keep close watch on your software certificate. Keep records about what controls you've signed, who signed them, and who had access to the signature key.
- Don't give your control privileges over your network.
- Set up your server so it knows who downloads the controls and when and where.

When deploying ActiveX controls in an intranet, consider the following:

- Code signatures do not necessarily mean full security.
- Don't accept HTML pages containing an ActiveX control from anyone.
- If you have a business partner that only publishes its data through an ActiveX control, allow only that ActiveX to run, and only in the context of the page it published.

Testing New ActiveX Controls

You can test a new control with the OLE control test container, which you start from the Tools menu in Developer Studio. After the container opens, choose Edit | Insert OLE Control to add the CIRCLE.OCX file to the container. You can manipulate the file within the test area.

This allows you to test the control before you put it into production. You can use a property bag, a stream, or storage to test the persistence of your control. You just put the control into the ActiveX Control Test Container and test it to make sure it works as the developer designed it. It is important to do unit testing when using ActiveX controls. Most errors can be detected before a control is used in production. Figure 7.7 illustrates an ActiveX Test Control Container.

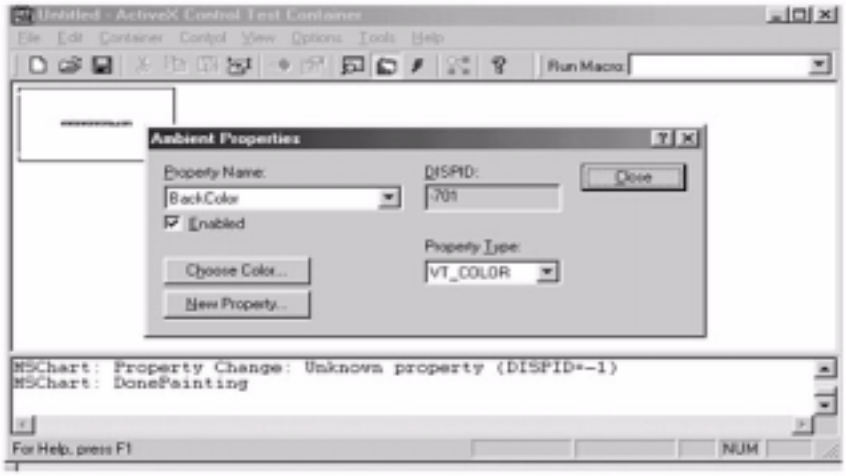

Figure 7.7 ActiveX Test Control Container.

It is important to remember that an ActiveX control is nothing more than a Dynamic Link Library (DLL). IE is an application that uses these special DLLs. Therefore, we can use the debugging technique previously described to debug our ActiveX controls.

Visual Basic

Visual Basic (VB) is a version of the BASIC programming language from Microsoft specialized for developing Windows applications. Dragging objects from the Visual Basic Toolbox onto the application form develops user interfaces. Visual Basic is widely used to write client front ends for client-server applications. As of Version 5.0, it is also used to create ActiveX controls for the Web (both EXEs and DLLs). Visual Basic for Applications (VBA) is a subset that provides a common macro language and is included with many Microsoft applications.

Visual Basic is similar to Java in that it is compiled into an intermediate language called *bytecode*. The bytecode is translated into x86 machine language by the Visual Basic run-time module. Starting with VB 5, native executable (.EXE, or execute) files can also be generated, but the run-time module, which provides necessary run-time functions, must still reside in the target computer.

Using Visual Basic for Testing

Visual Basic has many features that can support the testing process. For example, it has a host of intrinsic functions that can return important information about the test platform and the application under test (AUT). Visual Basic's Shell function and SendKeys statement can be used to run an application and manipulate its GUI. Visual Basic's Visual Database Tools will allow you to link to a database and view its data structure. You can get very sophisticated with Visual Basic and write essentially anything you need, such as a load testing application. Of course, the trade-off for a more sophisticated programming endeavor is that you will need the programmers and the time.

Visual Basic can also be used to test many behind-the-scenes operations of the application. For example, scripts can be written to access the .INI (initialization) files and the Windows Registry. Accessing the Windows API from Visual Basic is a very powerful way to

manipulate an application and return important information. This makes Visual Basic an ideal tool to use for testing. There are limitations, however, to using Visual Basic for testing. Visual Basic is not a test tool, so it doesn't include a lot of the extras that the high-end automation tools do. Visual Basic has no inherent support for bug reporting or test design and documentation, which many testing tools have. It also lacks a recording feature and any automatic test settings. If you want such features in Visual Basic code, your test team will have to write them. Therefore, Visual Basic is a good tool to do some basic testing but should not be used in high-volume and index testing.

COM Objects

A COM is a component software model that allows you to integrate services. It is a reusable component that is used for client-server applications. COM objects can be small or large, they can be written in several programming languages, such as Java, C + +, or Visual Basic, and they can perform any kind of processing. A program can call the object whenever it needs its services. Objects can be run remotely over the network in a *distributed object environment* (DCOM).

A COM object must be either a DLL or an EXE. The idea behind the COM is that the component can be reused in different applications that require similar functionality. You could create a tool for invoicing for one application and then use it in any other invoicing application you create or use.

Error handling should be coded into your application so that errors can be fixed and tracked. COMs can be set up different ways, so it is important that the tester evaluate the following:

- Is the application for a single user or multiple ones?
- Should the COM be set up as an EXE or DLL?
- Are all objects named properly?
- Are securities to the object set up properly?

Figure 7.8 illustrates the Visual Basic Environment when creating a COM object.

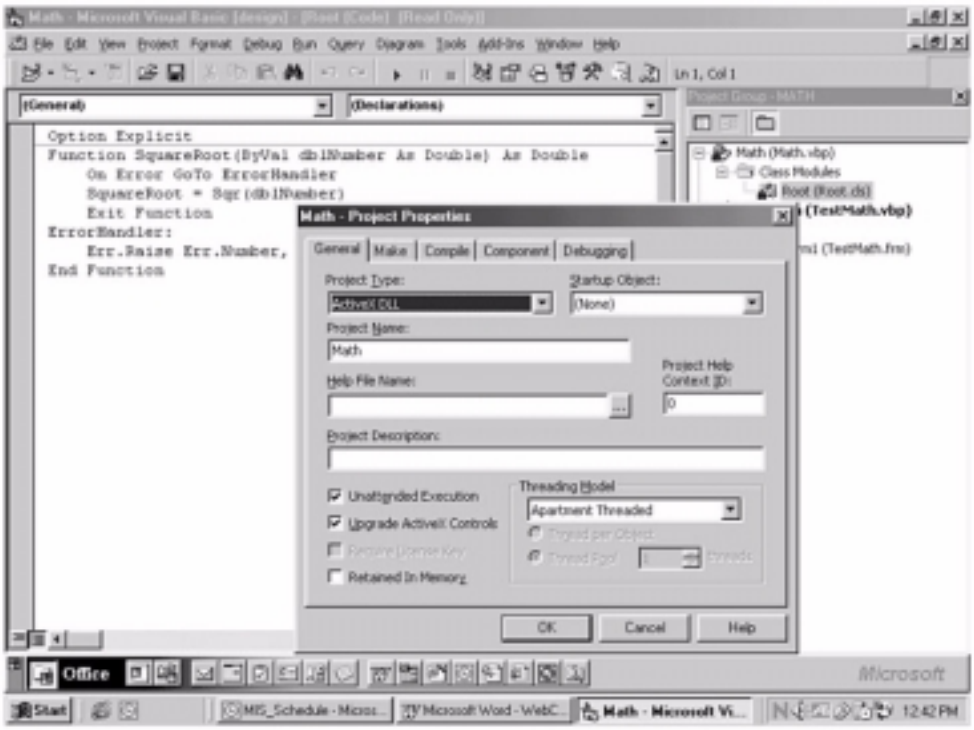

Figure 7.8 Creating a COM object.

Page 168

Automation (OLE Automation)

Standard applications, such as word processors and spreadsheets, can be written to expose their internal functions as COM objects, allowing them to be "automated" instead of manually selected from a menu. For example, a script could be written to extract data from a database, summarize and chart it in a spreadsheet, and place the results into a text document. This is why products like Visual Basic 6 have the capabilities to reference all the BackOffice products. To test for the OLE automation, you should first make sure that a reference exists. From this point on, error handling must be constantly used; for example, ensure that an object exists, check for errors in click events, and test for the CreateObject or GetObject functions.

Other Important Database and Security Features

Once you have created your database, chosen a scripting language, and decided if you will use ActiveX options or COM's, you are ready to address the security of the data that will be stored in your database. You need to set up accounts for your users that will include credit card and pricing information. The tester should know how the security is set up and should be aware of how the site will work. The following sections discuss a few of the items that will run off your Web site and will be stored in your databases.

Credit Card Transactions

There are several areas to consider when putting credit card capabilities on your site. You must make sure that your site is secure and transactions can be made without your customers worrying about security. Your testers should run several environment tests that will simulate all security issues.

Secure Sockets Layer

The leading security protocol on the Internet is Secure Sockets Layer (SSL). When an SSL session is started, the server sends its public key to the browser, which the browser uses to send a randomly generated secret key back to the server to have

a secret key exchange for that session. Developed by Netscape, SSL has been merged with other protocols and authentication methods by the Internet Engineering Task Force (IETF) into a new protocol known as Transport Layer Security (TLS). SSL is a protocol that is submitted to the World Wide Web Consortium (W3C) working group on security for consideration as a standard security approach for Web browsers and servers on the Internet. SSL can provide a security *handshake* that is used to initiate the TCP/IP connection. This handshake results in the client and server agreeing on the level of security that they will use, and this will fulfill any authentication requirements for the connection. SSL's role is to encrypt and decrypt the byte stream of the application protocol being used. This means that all the information in both the HTTP request and the HTTP response are fully encrypted, including the URL the client is requesting, any submitted form contents (such as credit card numbers), any HTTP access authorization information (user names and passwords), and all the data returned from the server to the client.

Transport Layer Security

The Transport Layer Security (TLS) is expected to become a major security standard on the Internet, eventually superseding SSL. TLS is backward compatible with SSL and uses Triple Data Encryption Standard (DES) encryption. To test your TLS you should use one of the tools that was mentioned in Chapter 5, "Web Site Testing Tools," to test for the security and the encryption. Figure 7.9 shows an example of security and how it reacts to a firewall and to encryption and decryption.

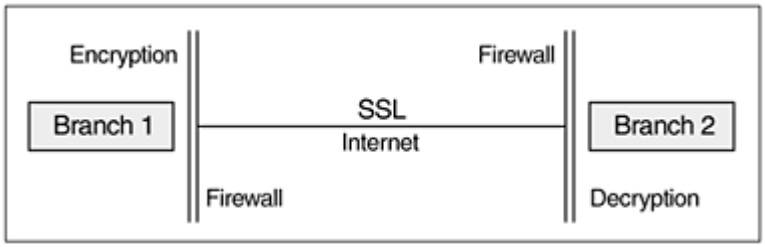

Figure 7.9 Security example.

Shopping Carts

A shopping cart is an online equivalent of the supermarket cart. You place your merchandise in the cart and then check out when you are all finished. There are several items that need to be tested when placing a shopping cart on your page. You must make sure that it gives you accurate prices and quantities and that you can add or delete items before you check out. Once again, you will want to refer to the test tools in Chapter 5 to ensure security and encryption. Shopping carts are considered temporary placeholders for items that will be

bought on the Internet. They can be persistent or nonpersistent. Persistent shopping carts will prevail even if you close your browser session. The shopping cart is implemented through cookies and references to the database.

Payment Transaction Security

Secure transactions create customer confidence. When customers purchase goods over the Internet, they can be apprehensive about giving credit card information; therefore, what your security measures are should be communicated to the customer. Figure 7.10 shows an example of a form that is used for payment when performing an order transaction.

Two things are needed to be tested to ensure that the customer's credit card information is safe. First, testing should ensure that the credit card information is transmitted and stored securely. Second, it should verify that strong encryption software is used to store the credit card information and that only limited, authorized access is allowed to this information.

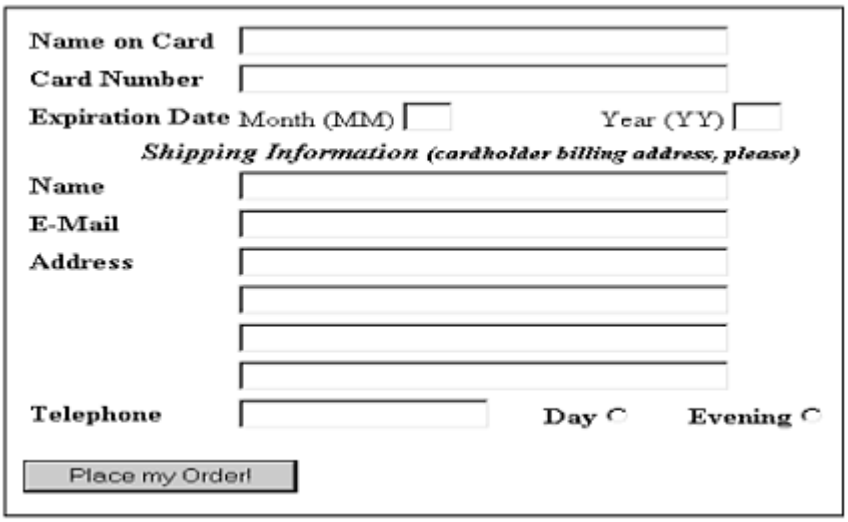

Figure 7.10 Example payment form.

Summary

As you can see, the Web technologies of HTML, Java, JavaScript, and VBScript are just a few of the Web tester concerns. It is a good idea for the tester to understand a variety of languages, networks, and databases.

Chapter 8 discusses how to test on different platforms and servers.

This page intentionally left blank.

CHAPTER 8

Testing on Different Platforms and Servers

Many problems that current Web sites face have nothing to do with development, but rather with deployment. The challenge of building Web sites with reliability, scalability, stability, and manageability needs to be addressed. As Web sites begin to handle more business-critical applications, the systems management and operational issues associated with Web development become crucial.

A new model for Web development has evolved to address these development and deployment issues. The Netscape Application Server has popularized Web application servers. The application servers form a clear level of separation between the Web server and data access layers. Web sites built using the application server model consist of at least three back-end layers:

- Web server
- Application server
- Data layers

The logic exists in the middle tier, with application servers handling all data manipulation and HTML page-creation functions.

The application server approach offers a number of natural advantages, particularly for applications that over time will grow in complexity in terms of business logic or number of users. Many Web applications need to interface with existing business systems, whether financial systems based on service access point (SAP) R/3 or PeopleSoft or transaction processing (TP) monitors, such as BEA Tuxedo. These products are able to integrate with such applications in ways that leverage the key advantages of their application server architectures.

Application servers are the backbone of enterprise computing on the Internet, but it has become increasingly difficult to weigh server choices. Keep in mind the bottom line is performance—if the server can't keep up with demand, your Web site is in trouble. The purpose of this chapter is to introduce you to the various types of available server platforms, server specifications, highlights, and features and to show how to set up and test applications on the Web server.

Web Servers

Web servers allow you to serve content over the Internet using HTML. The Web server accepts a request from browsers and returns the appropriate HTML documents. There have been a number of server-side technologies used to increase the power of the server beyond its ability to deliver standard HTML pages; these include CGI scripts, SSL security, and ASPs.

When testing the Web server, there are three important performance measurements:

- Response time
- Transaction rate
- Concurrency

The *response time* is the total time to send the request to the server and receive the complete response back. The response time is closest to the performance that the remote user sees. The lower the response time, the better the server.

The *transaction rate* is the total number of requests that can be processed per second. Because modern Web servers are multitasking and/or multithreading, it is possible for a Web server to support a much higher transaction rate than response time. The higher the transaction rate, the better the server.

To see how well the server multitasks among multiple simultaneous requests, a concurrency statistic measurement is used. *Concurrency* is when the average number of simultaneous connections to the server are fired at once. Unless the server is overloaded, this number will usually be close to the level specified by the requirements. The higher the number, the better the performance of the server because the system will not overload. An example is a script that allows for the program to change its threads six times to create six identical threads, each of which is running the retrieval tests concurrently. A switch can be set to the number of times each thread will try to retrieve a URL.

Metrics depend on more than just the Web server. They are affected by network bandwidth and latency, speed and amount of memory on the Web server host, and speed of the machine on which the test script is running. Table 8.1 lists the Web server testing features that should be tested.

Table 8.1 Web Server Testing Features

FEATURE	DEFINITION
Transactions	The number of times the test script requested the correct URL.
Elapsed time	The number of seconds it took to run the request.
Bytes transferred	The total number of bytes sent or received, less HTTP headers.
Response time	The average time it took for the server to respond to each individual request.
Transaction rate	The average number of transactions the server was able to handle per second.
Transferance	The average number of bytes transferred per second.

Concurrency	The average number of simultaneous connections the server was able to handle during the test session.
Status code nnn	This indicates how many times a particular HTTP status code was seen.

Choosing a Web Server

The choice of a Web server will depend on which platform you've selected to use and where your in-house expertise lies (knowledge of the OS or the programming that can extend your server's capabilities, for example). The versatility of TCP/IP networking means you can mix other server types in your intranet. You may be able to use a Unix server in a Windows NT network, but it may cause problems and may not be the best use of your resources. The platform and OS you choose for your Web server should be ones with which you are already familiar.

Installation Information

An understanding of TCP/IP and basic networking principles can facilitate a Web server installation. Several products are available that include online HTML-based documentation, which allows an easier installation process. Microsoft provides excellent HTML documentation. O'Reilly's print manual and HTML help are available. Apache provides almost no documentation other than some READ.ME files.

Once your server is running, you will need to make adjustments to it to meet your specifications. Some server packages allow more advanced customization than others; having access to the source code in Apache provides a great way to customize. More common is the ability to create custom error messages and default Web page headers and footers, something Enterprise, ICSS, and Web Commander can do.

Hands-on Attention

The nature of Web content means Web servers need hands-on attention. Thanks to Web browsers, you do not always have to be right at your machine to administer to your server. Most products on the market offer remote administration through a Web browser. IIS provides both a Windows utility and a browser-based administrator. Mac-based WebStar's remote administration, with its excellent use of color and space, is one of the best browser-based implementations on the market.

As Web servers have become more common in business, Web site management tools have kept up with the market and offer better resources. Site management tools are sophisticated enough to check and repair links and provide graphical maps of Web sites. Content management tools are becoming more

visually oriented, letting you create pages with drag-and-drop rather than hidden HTML tags.

Security

Because Web sites can share information, tight security and encryption have become important issues when choosing a server. The most common form of security supported by Web servers is basic authentication, in which users need to provide a user ID and password. Most servers support such basic authentication, but some servers go a step further and allow access restriction by IP address or host name.

Encryption can be used to protect against wire sniffers. Web servers use SSL to support encryption. All commercial servers support SSL, but some support more key-exchange and encryption algorithms. SSL creates a secure, encrypted channel between the server and browser by using certificate authentication. Using SSL, a certificate authority, such as VeriSign, provides server certificates for a fee.

Server Features

Web servers are no longer limited to transferring static HTML pages. Java and its supporting scripting languages provide an ideal Web development platform. Most servers support server-side Java. Several Web servers include custom APIs, and a few support the more common Netscape Server API (NSAPI). The most common form of Web programming uses a scripting language. Microsoft and Netscape even offer object-oriented, rapid applications development (RAD) tools for serious developers.

Database connectivity can be very important for companies that want to provide Web access to sales catalogs, parts databases, or legacy systems. Most products offer a range of data-access methods, from CGI code to ODBC calls through a scripting language. Most servers provide database access through an ODBC driver, although in some cases direct data access is faster.

Web Server Platforms

The following sections provide information that will help you decide which Web server is appropriate for your business needs.

Apache 1.1.3

Apache 1.1.3 offers a powerful and customizable approach for any Unix-based server. Experienced Unix users can enjoy the control they have over the Web server. You can download Apache and get all the Apache core and module source code, which can be modified to fit your needs. It may be difficult to use for newcomers to Unix because creating your own Web server by rewriting code can be difficult if you are not familiar with the Unix environment. Table 8.2 gives an analysis of Apache's features.

Apache will run on most Unix-based systems. Installing Apache is not an easy process because it requires manipulation before it will run. Apache does not have a management interface; the power is accessed directly through the text-based Unix shell. This can make Apache difficult to administer locally and may make it impossible to administer remotely

from a browser.

Apache can restrict access by IP addresses or by enabling basic security on individual directories. You can modify configuration files to implement either option. The public-domain version of Apache provides nothing beyond this basic level of security. If you require support for SSL encryption, you'll need the commercial version, Stronghold Apache, available from Community ConneXion.

Hosting multiple IP addresses on an Apache server is extremely simple. The HTTPD.CONF file contains two directives, Listen and VirtualHost, that enable Apache to perform multihoming with little configuration. The Listen directive tells Apache which port to monitor for a given IP address, and the VirtualHost

Table 8.2 Apache at a Glance

ISSUE	STRENGTH AND POWER	EASE OF USE	OVERALL RATING
Documentation	Poor	Fair	Fair
Installation/configuration	Good	Poor	Fair
Management/administration	Good	Poor	Fair
Content and site management	Not applicable	Not applicable	Not applicable
Security	Fair	Fair	Fair
Web development	Fair	Poor	Fair

directive contains the host information for the IP address. VirtualHost contains additional tags to provide Apache with a host name, root directory, server name, and separate logging information for the server.

There are some directives in Apache that provide basic site management capabilities, but there are no utilities included to handle content creation or management. The Apache's Redirect directive informs the server that a document has been relocated, either permanently or temporarily. URL mapping is handled in a similar fashion through the Alias directive. This directive will provide an alias in the root directory that will map to a document in a separate physical directory.

Unix developers may also like Apache's power and the ability to customize it. The more approachable Enterprise and Netscape FastTrack servers better suit Webmasters who aren't

Unix masters but are tied to the Unix platform nevertheless. You can find out more information about the Apache server at www8.zdnet.com/pcmag/features/webserver/iwsr1.htm.

Internet Connection Secure Server (WebSphere)

Webmasters of OS/2 sites who want to implement the most basic of Web servers might consider IBM's Internet Connection Secure Server (ICSS) 4.1. ICSS lacks site and content management tools and Java support, but IBM has released a newer version, which will add such features as an SNMP agent. Table 8.3 will show you some of the features of WebSphere at a glance.

Table 8.3 ICSS at a Glance

ISSUE	STRENGTH AND POWER	EASE OF USE	OVERALL RATING
Documentation	Fair	Good	Average
Installation/configuration	Good	Excellent	Above average
Management/administration	Fair	Fair	Fair
Content and site management	Not applicable	Not applicable	Not applicable
Security	Good	Good	Good
Web development	Fair	Fair	Fair

Versions of ICSS are available for AIX, HP-UX, Sun Solaris, and Windows NT Server and Workstation. ICSS offers versions for MVS/ESA and OS/400, but only the OS/2 version will be discussed here. ICSS is also available in nonsecure and export versions.

The installation of this product can take less than five minutes. Once it is installed, ICSS displays a default page showing you how to configure the server. Some documentation covering basic features is provided in HTML format. Webmasters can configure the server from any Web browser or by editing a single text configuration file. After configuration changes are made, the Web server must be restarted.

Most of the administration and configuration HTML forms are hard to comprehend and are not user friendly. Numbered lists display items that are available for editing, but you have to select the item by number from a separate drop-down list box that does not include the description. Other input fields can be scrolled but are hard to read because the print is small.

A big improvement over previous versions is the ability to host multiple Web sites on a single copy of ICSS. Redirecting URLs to other Web servers or even to other physical

directories on different servers is simple, as long as the OS/2 Warp Server itself is attached to the directory. The logging facility can be configured to display each log entry to the server window. You can open the server window on the OS/2 machine and view real-time statistics such as the number of transactions, the kilobytes sent, and the active connections.

Access control can be restricted to URL mappings by IP address, host name, user-defined protection setups, and access control lists (ACLs). ICSS can be installed on a machine that uses HPFS for its file system to support ACLs. The protection setups are confusing to define but are the only method of support for Secure HTTP (S-HTTP).

Although the secure version of ICSS 4.1 will support only SSL 2.0, version 4.2 does support SSL 3.0. Both versions will support standard encryption schemes and public and private key pairs. Like Enterprise, ICSS will maintain password and group files separately from the operating system. The IDs, passwords, and group memberships of users being added are kept in these two files. A user can belong only to one group.

ICSS handles enough connections to be acceptable for most commercial Web sites. It can process around 180 requests per second for static pages with one

processor and may strain under the heavy load of a corporate intranet. IBM is working to improve response time, adding multiprocessor capabilities and increasing the number of connections supported in version 4.2.

On the developer side, ICSS supports CGI, IBM's Internet Connection API (ICAPI), and SSI. ICSS does not directly support ODBC and SQL access to many popular databases; you should implement such support through ICAPI.

If you want to publish HTML documents on an intranet at your OS/2 site, or to publish securely over the Internet, ICSS may be for you. For advanced features, you may want to look at a more full-featured product. Version 4.2 merits a look as well.

The WebSphere Application Server is installed as follows:

1. Log into your machine with superuser (root) privileges.
2. If you have an earlier version of IBM HTTP Server than version 1.3.12 installed on your system, uninstall IBM HTTP Server. The installation of WebSphere Application Server will install version 1.3.12.
3. If a Web server on your system is running, stop it.
4. Run the install script file (./install.sh), which is in the /cdrom directory.
5. Click Next to pass the introductory page.
6. On the Install Options dialog, click Quick Installation and then Next.
7. On the Security Information dialog, fill in the user ID, password, and confirming password to use for the application server and click Next.

8. Specify the destination directory.
9. If you do not have IBM HTTP Server already installed on your system, a fixed destination directory (/usr/HTTPServer) for IBM HTTP Server will be shown.
10. Click Next.
11. Click Next again and then OK to begin the installation.

1. The next page points you to the README. If you do not use a Netscape browser or if the installation program cannot open a browser, look in the &main_Application_Server_directory>/Web/InfoCenter/was directory for the README.HTML file.
2. For the most recent version of the README or release notes, go to Library section of the product Web site at www-4.ibm.com/software/.
3. Click Finish.

You can find out more information about the Internet ConnectionSecure server at www8.zdnet.com/pcmag/features/webserver/iwsr2.htm.

Lotus Domino

Lotus Domino 4.5a, Lotus Development Corp.'s most recent Notes server release, is more than a Web server. Domino now integrates Web server capabilities and builds on the mail, scheduling, and groupware applications, which are features of its predecessors. Those who are already using Notes and those who want the groupware capabilities and dynamic page generation Domino provides should consider using Domino as their Web server. Table 8.4 gives an analysis of its features.

Domino is able to publish information stored in Notes databases over the Web. Whereas most Web servers handle requests for Web pages as simple file transfers,

Table 8.4 Lotus Domino at a Glance

ISSUE	STRENGTH AND POWER	EASE OF USE	OVERALL RATING
Documentation	Excellent	Good	Above average
Installation/configuration	Excellent	Good	Above average
Management/administration	Excellent	Excellent	Excellent
Content and site management	Excellent	Good	Above average

Security	Good	Good	Good
Web development	Excellent	Good	Above average

Domino has to add necessary HTML wrappers and perform image conversions and other Notes-to-Web conversions on the fly. Domino caches only images.

Domino can handle about 200 static page requests per second. Email, discussion groups, project tracking, and customized workflow applications are all stored in Notes databases. Domino lets you access these applications via a standard Web browser instead of regular Notes client software, so you can extend the power of your groupware. If you already have Notes in place, this feature makes it easy to consolidate applications development to service both Notes and Web users. A Domino server is a great way to push all your groupware applications to the Web.

The administration features in Domino are solid. You can add new users using the Notes Public Address Book. It is easy to add new users and groups and assign privileges to different Notes databases. The Windows NT version of Domino is integrated with Windows NT's User Manager, so entries added to Domino can be copied to User Manager and vice versa.

Domino comes with a separate configuration database from which you can configure multiple host names and map logical URLs to physical directories. Domino creates standard HTTP-access and error-reporting logs. Domino can be configured to print logs either as a text file or to a separate Notes database.

Domino continues Lotus Notes' tradition of strong data security. Although it doesn't support SSL 3.0, Domino can self-sign digital certificates, which is useful for test environments in which it isn't necessary to use a third-party certificate authority. Domino can act as a local certificate authority and certify key pairs generated by other servers within an intranet environment. The key-generation and certification process can be confusing for most people the first time around, but Domino provides excellent step-by-step documentation.

A limiting aspect of Domino is that you cannot place security restrictions on flat files stored outside of Notes databases. Domino can serve static HTML pages and execute CGI scripts, although the Web development environment in Domino revolves around Notes databases, and Lotus recommends that administrators store Web information in Notes databases rather than in static files.

Domino will also allow you to make a full-text index of Notes databases, so you can incorporate text searches from either a Web browser or a standard Notes client.

Most form elements in Notes, such as binary attachments, graphics, and tables, usually map correctly to HTML, but other form elements, such as buttons, don't map smoothly to HTML constructs. Domino's documentation does a good job of outlining which Notes design

features are supported and provides recommendations for working around limitations in HTML. In short, if your primary concern is creating the kind of collaborative applications that Notes traditionally provides, Domino is a great tool for moving these applications to the Web.

The Domino server Incremental Installers now require two files for Windows 32 and OS/2 platforms. Following are instructions for downloading and launching the Incremental Installers.

NOTE After you download the files, run each file to decompress it automatically. This "unzips" the file to its full size. Once you have done this, you are ready to follow the steps to complete the installation.

Download Instructions for Windows 32 (Server)

1. Download each file (W32NSRVG and DOLS_W32N). Place the dols_w32n file in your data directory.
2. Launch the self-extracting executable with the Notes Database icon first (W32NSRVG).
3. At the command line prompt, change into the Notes/Data directory.
4. Launch the corresponding Domino Off-Line Services executable (DOLS_W32N).

If you are using a partition server, run the preceding steps from each data directory.

Download Instructions for the OS/2 Platform

1. Download each file (OS2SRVG and DOLS_OS2). Place the dols_os2 file in your data directory.

1. Launch the self-extracting executable (OS2SRVG).
2. At the command line prompt, change into the Notes/Data directory.
3. Launch the corresponding Domino Off-Line Services executable (DOLS_OS2) with the command line option -d (for example, DOLS_OS2 -d). If you are asked if you would like to overwrite the directory, select "All." This will place files in and below your Data directory. (If you are using a partition server, run the preceding steps from each data directory.) You can find more information about Lotus Domino at http://www8.zdnet.com/pcmag/features/webserver/iwsr3.htm.

Luckman's Web Commander

NOTE Even though Luckman's Web Commander is no longer a viable Web server

solution (the company has gone out of business), you can still take away valuable information from the discussion presented in this section.

Luckman's Web Commander had a hard time handling static page requests during testing, but its interesting mix of features and simple maintenance requirements made it feasible for the small to medium-size Web sites for which it was designed. Table 8.5 gives an analysis of its features.

Web Commander included a Web server, an HTML editor, and development tools. It also included a copy of Seattle Lab's mail server, SLMail SMTP/POP3 mail server, and WebCharge, an automated credit card verification and clearing program for commercial sites.

Table 8.5 Luckman's Web Commander at a Glance

ISSUE	STRENGTH AND POWER	EASE OF USE	OVERALL RATING
Documentation	Good	Fair	Fair
Installation/configuration	Good	Fair	Fair
Management/administration	Good	Good	Good
Content and site management	Fair	Good	Fair
Security	Good	Excellent	Above averge
Web development	Fair	Fair	Fair

The first time you ran Web Commander, a configuration wizard walked you through the setup. Web Commander's configuration interface was comprehensive, providing clear menu choices and helpful instruction messages. One interesting feature was the program's ability to save each virtual server's configuration data to a file. This provided a backup in case of a disk crash or operator error. There was no remote administration feature, so all administration had to be done at the server.

Web Commander provided several types of security. Access could be restricted based on an IP address or a domain name. You could also require each user to have a unique user name and password. Web Commander did not use the Windows NT user database, so you needed to manually create an account for each user. Web Commander did not support SSL 3.0, but it provided S-HTTP as well as a security wizard to help you order a security certificate by email from Verisign or any certificate authority.

A unique feature of Web Commander's was WebCharge, a credit card verification

program. It accepted credit card information from a user and processed the transaction through one of several online credit clearing services. When used with SSL or S-HTTP, WebCharge provided a secure way to conduct business over the Web.

Luckman provided several HTML development tools with Web Commander, including WebPage, an automated, wizard-based page generator; WebStudio, a code-based HTML editor; and WebMap, an image map. There were some problems with the tools interacting with each other, but Luckman eventually acquired WebEdit, a full-featured, code-based HTML editor designed to solve some of the issues that surround WebStudio. The Web Commander also included kits for developing applications in Perl and Java, with a copy of the Sun JDK and Microsoft's Perl CGI development kit. An ODBC driver and example applications allowed access to SQL databases.

Two search engines were included with Web Commander. The Windows 95-based servers used the Wide Area Information Server (WAIS), and Windows NT servers used the Excite search engine. WAIS, which performed only text-file searches, was more limited than Excite and required knowledge of the Perl programming language. Excite was faster and searched both text and HTML documents. Because the files on the Luckman CD were stored in a proprietary format, you had to download and install Excite from the Excite Web site.

Web Commander was a capable package with plenty of promise. Luckman had planned to announced a 2.0 release of Web Commander, with remote administration, a proxy server, and better authoring tools.

Microsoft Internet Information Server

The Microsoft Internet Information Server 3.0 (IIS 3.0) is a comprehensive solution that will help Windows NT replace some of the Unix-dominated Web servers. IIS 3.0 comes with extras that include Active Server Pages for building dynamic Web pages, Crystal Reports for custom reporting, Microsoft FrontPage 97 for site management, Index Server for advanced searching, and NetShow for on-demand multimedia. Table 8.6 gives an analysis of IIS' features.

IIS uses Windows NT's User Manager to maintain users and groups, which will save the trouble of maintaining multiple sets of network and Web site users. IIS uses Windows NT's Event Viewer and Performance Monitor to view such items as bytes sent per second and current CGI requests.

IIS can handle Internet Server API (ISAPI) processing on the server side. IIS began to overtake the competitors as the load was increased to 56 and then 60 clients with static pages, boding well for its scalability. IIS also was the top performer on our latency tests.

Table 8.6 Microsoft Internet Information Server at a Glance

ISSUE	STRENGTH AND POWER	EASE OF USE	OVERALL RATING

Documentation	Excellent	Good	Above average
Installation/configuration	Good	Excellent	Above average
Management/administration	Excellent	Excellent	Excellent
Content and site management	Good	Excellent	Above average
Security	Good	Excellent	Above average
Web development	Excellent	Excellent	Excellent

IIS comes with three default services:

- WWW
- FTP
- Gopher

The Internet Service Manager (ISM) application controls the services on this and any other IIS server on the network. ISM is run from the Windows NT Server, a Windows NT, or a Windows 95 workstation. For remote administration, you can run an HTML version of ISM from a browser.

As a Web developer, you may want to consider IIS and Active Server Pages (ASP). Keep in mind that ASP provides an extensive server-side platform supporting compile-free, language-independent scripts and ActiveX components. IIS returns all ASP requests as standard HTML and lets you create dynamic Web sites and online applications that are accessible by any browser. IIS also supports Java through a Java virtual machine.

Database access is extended in this version of IIS. ActiveX Data Objects (ADOs), an ASP component, lets developers' access and control data in any ODBC- or OLE DB-compliant database using any ActiveX scripting language. Developers can put a Web front end on almost any legacy database without arcane CGI programming. This makes it easier for developers to maintain databases.

To use the power of ASP and server-side scripting, IIS includes native scripting engines for VBScript and JScript. Server plug-ins are available for other scripting languages such as Perl, TCL, and REXX. You can even use several different scripting languages within a single ASP document. IIS also allows you to report on the use of the Web because it comes with Seagate Software's Crystal Reports 4.5 for IIS. The resulting presentation-quality reports consist of log-file data returned in standard HTML format, complete with graphs and tables. The reports can be customized to meet the needs of your business.

IIS has done well with content and site management by way of the FrontPage. The simple-to-use FrontPage editor lets you build complex Web pages and Active Server Pages without having to write a single line of HTML code. Other items that come with IIS include Index Server 1.1 and NetShow. Index Server

allows you to index and search site content and perform advanced searches on document properties. The NetShow add-in allows you to deliver audio, video, text, and images to users on low-bandwidth networks using multicasting and data-streaming techniques.

Building on Windows NT's security prowess, IIS provides additional levels of security. Access can be restricted to a directory or URL by user, group, or IP address or by using Windows NT's Challenge/Response authentication or SSL 3.0. But unlike Netscape Enterprise Server, IIS cannot limit access by host or domain name.

The IIS 3.0 shows that being a Webmaster no longer means you have to be a Unix expert or CGI programmer. Whether you are implementing a small intranet or a large Internet site on a Windows NT platform, this package may be the Web server for you. You can find out more information about the Microsoft Internet Information Server at http://www8.zdnet.com/pcmag/features/webserver/iwsr5.htm.

Netscape Enterprise Server

The Enterprise server includes site and content management tools and incorporates a solid Web development platform. This server is for medium- to large-size Web sites that require sophisticated control over the server and site content. Enterprise performed well in testing and, under Windows NT, it even outdid some of its Unix brethren. Table 8.7 gives an analysis of its features.

Table 8.7 Netscape Enterprise Server at a Glance

ISSUE	STRENGTH AND POWER	EASE OF USE	OVERALL RATING
Documentation	Excellent	Good	Above average
Installation/configuration	Excellent	Excellent	Excellent
Management/administration	Excellent	Fair	Average
Content and site management	Excellent	Excellent	Excellent
Security	Good	Good	Good
Web development	Excellent	Excellent	Excellent

Once you go through the installation, you can tweak Enterprise in many ways. Because it builds on its Unix roots, you can control everything from performance tuning to creation of default configuration styles for documents. The management interface runs from any browser that will support frames and JavaScript, letting you manage the server remotely from any client or platform.

Enterprise provides options for mapping and forwarding URLs and supports URL redirection to directories on other servers. For large Web sites, Enterprise supports hardware and software virtual servers. Hardware virtual servers must share the same server configuration but can have different IP addresses; software virtual servers must share an IP address but can have different configurations. To host multiple virtual servers with different addresses and configurations, you need to install a separate instance of Enterprise for each virtual server.

Enterprise supports everything from CGI to JavaScript and NSAPI. The LiveWire module comes with Site Manager and Application Manager. The Site Manager resembles Microsoft FrontPage 97, with tools for tracking all site elements and verifying internal and external hyperlinks. It also includes several Web site templates for creating a complete site from scratch.

The Application Manager allows you to create, modify, debug, and run a Web application using RAD-like tools. The Debug option is especially useful for tracing and debugging script code. Using LiveWire and Navigator Gold, you can create and organize a site, develop and compile scripts, and deploy the application to any destination directory on your server.

The Enterprise comes with a JavaScript Guide to help you start writing client-side scripts with Netscape Navigator. JavaScript allows you to automate elements on your pages and make them truly dynamic. On the server side, additional JavaScript capabilities let you develop full-blown client-server applications. For database connectivity, LiveWire gives JavaScript methods for accessing various data types using ODBC. For a more direct approach, you can access Informix, Oracle, and Sybase databases without using ODBC.

The Enterprise supports encryption and SSL to help eliminate eavesdroppers and intruders. Using the Restrict Access option, you can limit access by user, group, IP address, or host or domain name. To increase security, you can enable stronger DES or RC4 ciphers when using encryption. Enterprise maintains users

and groups in a .DBM file that is separate from the Windows NT domain. To add a user to multiple groups, you need to add the user and then go back and edit the user's profile.

Netscape Enterprise Server has a lot to offer, and it meets the needs of medium- and large-scale Web sites. You can find out more information on Netscape Enterprise Server at

http://www8.zdnet.com/pcmag/features/webserver/iwsr6.htm.

Netscape FastTrack Server

Netscape FastTrack is a scalable cross-platform Web server suitable for an intranet workgroup setting or a mid-size Web site. Even though FastTrack lacks the integrated development tools of some other products, it does provide open development interfaces that make it appropriate for more advanced Web applications. For more sophisticated options, such as text searching, advanced site management, and document versioning, you'll have to purchase an add-on or look to a higher-end product. Table 8.8 gives an analysis of FastTrack's features.

FastTrack's installation and configuration processes are easy and are similar across platforms. FastTrack will run on a number of Unix platforms, as well as on Windows 95 and NT. To install FastTrack under Windows NT, run a setup utility and answer a few basic questions. During the installation, the utility configures a special HTTP server that is used to configure and manage additional FastTrack Web server processes. Administrators can create Web servers by accessing this server port using Navigator Gold.

Table 8.8 Netscape FastTrack Server at a Glance

ISSUE	STRENGTH AND POWER	EASE OF USE	OVERALL RATING
Documentation	Excellent	Excellent	Excellent
Installation/configuration	Excellent	Excellent	Excellent
Management/administration	Excellent	Fair	Average
Content and site management	Good	Good	Good
Security	Good	Good	Good
Web development	Good	Not applicable	Good

After the initial setup, accessing a Web application called Netscape Server Manager (NSM) that comes from Navigator Gold configures FastTrack. On the positive side, the remote administration is easy. You can specify the remote IP addresses or host names from which FastTrack will accept requests to modify server parameters. On the negative side, the Web-based interfaces aren't as flexible as native GUI interfaces.

When you get used to the interface, you'll appreciate FastTrack's configuration options. Small ISPs that use FastTrack to provide multiple domains will welcome the support of virtual servers. The virtual servers will let you map multiple host names either to a single IP address or to multiple addresses. Unix requires more tweaking than Windows NT when dealing with more than one IP address, so before configuring any virtual servers, you should make sure the network interface card and driver are fully configured for multiple TCP/IP addresses.

FastTrack maintains its own user database separate from the operating system, so adding users to the system is not as easy as it is using other servers, but having a separate user database does provide you with control over each user and group.

FastTrack's security is comprehensive; it supports all six ciphers in SSL and Challenge/Response Authentication. As an additional client-side security measure, FastTrack supports a new HTML directive that prevents caching preencrypted files to the client. Netscape provides a utility to generate the public-key/private-key pair. The key-generation process is easy, and the documentation's description of Web server security issues is helpful. To assist in load balancing and security, FastTrack supports URL redirection and forwarding, eliminating the need to change all the links in each HTML document as your site grows and changes.

FastTrack ranked among the fastest performers, scoring just a notch below the fastest performer, IIS. The Digital Unix version was a solid performer. For static files, the Digital Unix and Windows NT versions of FastTrack performed almost identically, with performance increasing steadily through 40 clients before leveling off. The SGI version of FastTrack leveled off at 20 clients, and at high client loads performed about half as well as the Digital Unix and Windows NT versions. Adding a single processor resulted in considerable performance gains, regardless of platform. When a processor was added to the underlying hard-

ware for each platform, performance gains of about 50 percent on each of the three tests occurred.

FastTrack offers features for Web developers. At the most basic level, FastTrack can execute CGI scripts, WinCGI, and Server Side Includes (SSI), which are the industry-standard protocols for implementing server-side programming logic. FastTrack includes a server-side Java interpreter to let developers write Java applications that launch from the Web server in response to Web requests. C programmers who need flexibility from a Web server can use NSAPI to modify FastTrack's core functionality. You can find out more information about Netscape FastTrack Server at http://www8.zdnet.com/pcmag/features/webserver/iwsr7.htm.

Novell Web Server

Novell Web Server (NWS) has an integrated search engine, SSL 3.0 support, and enhanced database connectivity; Novell's new Web server can be used for corporate intranets. Table 8.9 gives an analysis of its features.

The luxury of NWS is that you can use your existing NetWare server and avoid the expense of a separate Web server. The minimum NetWare configuration with NWS needs only 32 MB of RAM, but you may want to add memory to your server for better performance. Just configure your server for TCP/IP and use NetWare's INSTALL.NLM to load the NWS modules; the NetWare server will transform into an intranet server in minutes.

Table 8.9 Novell Web Server at a Glance

ISSUE	STRENGTH AND POWER	EASE OF USE	OVERALL RATING
Documentation	Good	Fair	Fair
Installation/configuration	Good	Excellent	Above average
Management/administration	Excellent	Good	Above average
Content and site management	Good	Good	Good
Security	Good	Good	Good
Web development	Good	Fair	Fair

NWS processes about 800 requests per second for static HTML, landing just slightly behind the pack leaders. NWS integrates fully with NetWare Directory Services (NDS), which eases the administrative burden by centralizing the management of users, groups, and NDS objects under one NDS management umbrella. A convenient NDS browser utility allows you to view all objects in the NDS tree from a standard Web browser. You can control intranet access for members of the NDS tree using familiar NetWare rights and restrictions. For users outside the NDS tree, you can restrict access by IP address, physical network address, or domain name.

Although NWS supports traditional URL redirection and forwarding, you can now store Web pages on any NetWare server on your network. This allows you to distribute your intranet resources among several servers. You can also control all aspects of any NWS-enabled Web server on the network using the WEBMGR.NLM management module.

For development, NWS supports applets written in Java. For more experienced developers, NWS provides its own kind of CGI, called LCGI (Local CGI). Perl developers will find the Novell's Perl 5 script interpreter easy to use. For advanced scripting, the NWS comes with NetBasic. If you need to provide access to your Oracle database via the Web, NetBasic includes function classes for connecting directly to Oracle 7 databases.

The QuickFinder is a full-featured search engine that lets you add search capabilities to your intranet easily. You can use the power of the search engine in your Web applications using the QuickFinder API in conjunction with NetBasic. Although NWS is the sole NetWare product for Web servers, it provides a simple way to leverage the strengths of NDS on the Web using existing NetWare servers. You can find out more information about Novell Web Server at http://www8.zdnet.com/pcmag/features/webserver/iwsr8.htm.

WebSite Professional

WebSite Professional is one of those special toolkits; it has most everything you may need packaged in the product. Table 8.10 gives an analysis of its features.

WebSite runs under Windows 95 or Windows NT as either a service or an application. Installation takes only a few minutes. The installer program installs WebSite's server and utilities, Sausage Software's Hotdog HTML editor, and Allaire's Cold Fusion Web applications development tool. The wizard-based

Table 8.10 WebSite Professional at a Glance

ISSUE	STRENGTH AND POWER	EASE OF USE	OVERALL RATING
Documentation	Excellent	Excellent	Excellent
Installation/configuration	Excellent	Excellent	Excellent
Management/administration	Good	Good	Good
Content and site management	Excellent	Excellent	Excellent
Security	Good	Excellent	Above average
Web development	Good	Excellent	Above average

setup makes setting up multiple virtual servers easy. It allows you to map any URL to any virtual server and can even map URLs to another physical machine. These features make WebSite a nice choice for ISPs and for any company needing to host multiple Web sites on a single machine.

If you want to administer a WebSite server remotely, you should install a customized copy of WebSite on the remote machine. It lets you restrict access based on a user's IP address or domain name. It also allows you to require each user to have a unique account name and password and maintains its own user list that is separate from the Windows NT

user database. This means you must manually create user accounts for everyone who will have access to the server. Once you've done that, you can use WebSite's flexible user and group security features to control read and write access to each URL on the server. Users with write access can use an HTTP 1.1 PUT browser to post updated pages as needed. If you want to be informed of every detail of your server's operation, the WebSite's comprehensive logging and tracing features are thorough. WebSite maintains three separate log files. They are the access, server, and error logs that are in either National Center for Supercomputing Applications (NCSA) standard or WebSite extended format. The trace option lets you log 11 different server activities selectively.

WebSite offers Web developers programming interfaces, including CGI, ISAPI, server-side Java, Visual Basic, and O'Reilly's own WebSite API (WSAPI). The Cold Fusion Web applications development tool that comes with WebSite provides on-the-fly access to ODBC and SQL databases. This makes it easy to

create front ends to most databases. O'Reilly provides software development kits for WSAPI and Cold Fusion, as well as a sample Web site with working examples of all the programming interfaces. The well-documented example programs include links to additional programming resources on O'Reilly's own Web server.

WebSite includes content-creation tools that don't require any programming. The indexing tool WebIndex builds keyword index files that can be searched with the companion CGI program, WebFind. You can create the index with WebIndex.

WebSite delivers a complete suite of tools to build and manage a Web site. WebSite performed well when delivering static HTML pages and acceptably when CGIs were added into the mix of page requests. It is one of the most complete, capable, and usable Web server packages on the market. Its completeness, flexibility, and excellent management features make it a reasonable choice for corporate intranets and ISPs. You can find out more information about WebSite Professional Server at http://www8.zdnet.com/pcmag/features/webserver/iwsr9.htm.

StarNine's WebStar

Turning your Mac into a Web server can be done with StarNine's WebStar. The WebStar adds performance and features that may take it well beyond its predecessors. Server administration can be handled from the server console or remotely via any Web browser. The administration utility allows you to monitor server activity and maintain site security, default server settings, and plug-ins. WebStar's browser administrator makes excellent use of screen space, letting you navigate easily through available functions. Table 8.11 gives an analysis of its features.

Security within WebStar is simple; you can quickly set up security parameters, known as *Realms,* and assign users to them. Existing lists of users can be imported into WebStar directly. An Allow/Deny feature lets you restrict access by IP address or domain name. It also allows you to apply restrictions to individual Realms. WebStar/SSL ships as part of the basic package for security. The SSL server is a separate application but runs simultaneously with the primary Web server. This allows secure and nonsecure access to your site.

Table 8.11 WebStar 2.0.2 at a Glance

ISSUE	STRENGTH AND POWER	EASE OF USE	OVERALL RATING
Documentation	Good	Excellent	Above average
Installation/configuration	Good	Excellent	Above average
Management/administration	Good	Excellent	Above average
Content and site management	Fair	Good	Fair
Security	Good	Excellent	Above average
Web development	Good	Good	Good

You can build, extend, and maintain your Web site with the tools included in the package. StarNine provides a copy of Adobe PageMill for the easy creation and maintenance of Web sites. WebStar includes support for server-side Java, SSI, and CGI, as well as the unique WebStar API (W*API). Installing plug-ins is as simple as dragging and dropping them into WebStar's plug-ins folder and restarting the server.

The Mac OS is unable to handle multiple IP addresses; WebStar's multihoming capability is limited. To remedy this StarNine suggests routing the multiple hosts to a single IP address. Then specify a unique home page for each host, placing the documents for each host in unique trees or running multiple copies of the server, each with its own configuration.

In a market dominated by Unix and Windows NT, StarNine has provided Macintosh users with the power to command a segment of the Internet. If you are accustomed to working on the Macintosh platform, WebStar can be a valuable addition to your Internet toolkit. You can find out more information about WebStar at http://www8.zdnet.com/pcmag/features/webserver/iwsr10.htm.

Service Problems

Once you have your server in place, you may find that certain items will not work the way you expect them to. You will want to be able to troubleshoot the problem before you contact technical support. Table 8.12 lists basic server problems.

Table 8.12 Common WWW Troubleshooting Service Problems

PROBLEM	SUGGESTION	ADDITIONAL INFORMATION
Web server cannot be found.	There may not be a DNS entry; the IIS computer must have a domain name entry in the DNS server responsible for your network.	You may want to use the command line NSLOOKUP.EXE tool to check for name records for any machine on the Internet.
The Virtual Website cannot be created.	There needs to be a unique port number, IP address, or host header for the Web site.	Search for the online documentation for creating Virtual Websites.
Cannot require SSL.	The Certificate must be installed to require SSL for the Web site.	Search for online documentation for the SSL.
Another IP address cannot be selected.	The Windows NT Server should be configured to respond to additional IP addresses.	The configuration is done through the Network control panel.
The server cannot be found by other name.	For Web browsers to find your site, make sure your IIS computer has an entry corresponding to that name in the DNS server responsible for the network.	Search for online documentation for DNS server entries.
The browser cannot find the virtual site.	If host headers are being used, the browser should support host headers or the CGI/ISAPI workaround.	If a port is being used other than 80 on the site, any reference to the site must explicitly reference the port in the URL.
Browsers with plug-ins for multimedia files ask if you want to save the disk rather than displaying the data.	A MIME type has to be defined for data types other than those already defined in the IIS setup.	This problem may also be associated with other types of data.
The site has moved and the browser cannot find it.	The browser needs to be redirected to the new location of the Web site.	The URL option in the Home Directory tab of the Web site's property sheets allows you to redirect the site.

Table 8.12 (*Continued*)

PROBLEM	SUGGESTION	ADDITIONAL INFORMATION
The users have their access denied after they enter an account name and password.	An account must be defined for the users if the Anonymous Authentication is disabled.	Find online documentation for Anonymous Authentication.
You are unable to log on to the IIS Administrative Web site.	The Windows NT Challenge/Response should be enabled to use the administrative Web pages.	Search for online documentation for the Windows NT Challenge/Response.
Anonymous user cannot access any files.	The anonymous account that is defined in the Web site Authentication Properties sheet must also exist and have the same password as a Windows NT account.	Search for documentation for Web site Authentication Properties sheet.
The only browser that can be authenticated by your Web site is the Internet Explorer.	Web browsers other than IE requires Basic Authentication to be authenticated as anything other than an anonymous user.	Search for online documentation on Basic Authentication.
User cannot access the Web site data that is stored on the Universal Naming Convention or Uniform Naming Convention (UNC) share.	Share permissions need to be set correctly.	An account name and password need to be established for Web site access to share.

Summary

The information that was included in this chapter came from different articles on the ZD Net Web site. As you can see, a lot of time and money was spent testing and exploring different servers. Even though some of this information may be dated by the time this book goes to press, it should give you an idea of the importance of servers and how they will affect the Web market. It is important to know the needs of your business before investing in a Web server.

Chapter 9 discusses Web capacity testing.

This page intentionally left blank.

CHAPTER 9
Web Capacity Testing– Load and Stress

Load and stress testing are critical components of Web testing. This type of testing requires many simultaneous users to make requests during peak activity that will put a large load on the Web server's processor. The key to a successful Web site is to have the hardware configured correctly so that it will be powerful enough to meet the demands required. Load and stress testing are essential to ensure that these demands are met. By performing load testing, you will be able to find performance bottlenecks in your design and setup during the early stages of development. Figure 9.1 illustrates a basic setup for a load testing environment, and Figure 9.2 illustrates how the load testing will be set up for the Web test.

The performance of the load or stress test Web site should be monitored with the following in mind:

- The load test should be able to support all browsers.
- The load test should be able to support all Web servers.
- The tool should be able to simulate up to 500 users or playback machines.

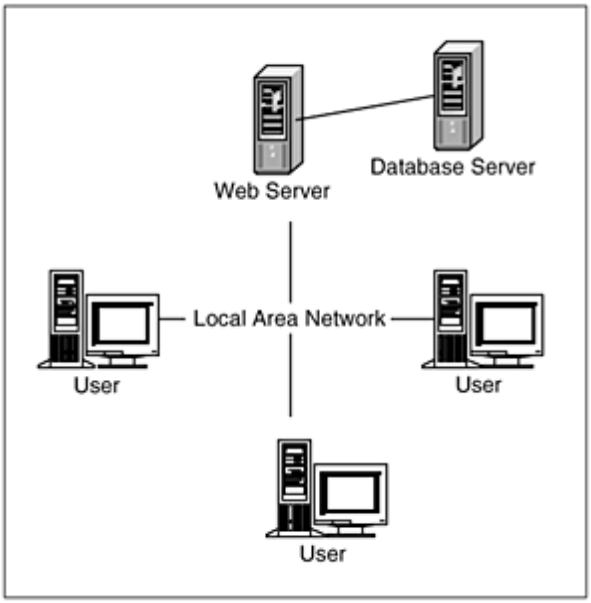

Figure 9.1 Basic setup for a load testing environment.

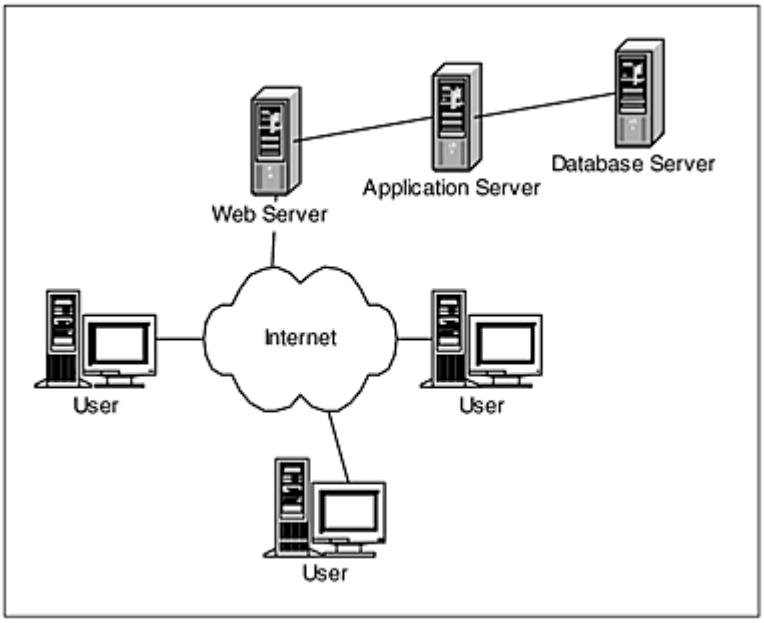

Figure 9.2 Load testing setup for the Web test.

- The tool should be able to run on Windows NT, Linux, Solaris, and most Unix variants.
- There should be a way to simulate various users at different connection speeds.
- After the tests are run, you should be able to report the transactions, URL, and number of users who visited the site.
- The test cases should be assembled in a like fashion to set up test suites.
- There should be a way to test the different servers and port addresses.
- There should be a way to account for the user's cookies.
- As mentioned in Chapter 7, "Testing Languages and Databases," there should be a way to test for the back-end process, including Active Server Pages, applets, servlets, plug-ins, ActiveX components, ISAPI, and cgi-bin.

Load Testing

Load testing is a simulation of how a browser will respond to intense use by many individuals. The Web sessions can be recorded live and set up so that the test can be run during peak times and also during slow times. The following are two different types of load tests:

Single session. A single session should be set up on a browser that will have one or multiple responses. The timing of the data should be put in a file. After the test, you can set up a separate file for report analysis.

Multiple session. A multiple session should be developed on multiple browsers with one or multiple responses. The multivariate statistical methods may be needed for a complex but general performance model.

When performing stress testing, looping transactions back on themselves so that the system stresses itself simulates stress loads and may be useful for finding synchronization problems and timing bugs, Web priority problems, memory bugs, and Windows problems using API. For example, you may want to simulate an incoming message that is then put out on a looped-back line; this in turn

will generate another incoming message. Then you can use another system of comparable size to create the stress load.

Memory leaks are often found under stress testing. A *memory leak* occurs when a test leaves allocated memory behind and does not correctly return the memory to the memory allocation scheme. The test seems to run correctly, but after several iterations, available memory is reduced until the system fails.

Load Testing Scripts

Some of the following information about load testing and other components of Web testing came from the Segue company; other information on load testing for the Internet came from Stefan Asbock. In his book, *Load Testing for eConfidence* (Segue Software, 1992–2000), he provides information about how to understand and work with load testing and setting up load testing scripts. According to Asbock, a load testing script is "a specific set of operations for a virtual user to perform, designed to replicate real-world user activity on a Web application."

A script is a language based on an extension of another language. This script should be developed to call various scenarios developed from the business requirements and allow the tester to communicate with the Web server via the Web application. Asbock continues:

> A transaction is the basic component of a load testing script; it is a sequence of Web application accesses, involving user interaction and having a certain logical sequence.
>
> For example:

- A user sends several transactions setting up a request.
- The user waits to receive the response from the request.
- The user will provide interaction to the system through the request.
- And this cycle is repeated over and over until the test is complete.

Asbock uses the following example:

> A transaction might consist of the sequence of events where a user accesses an ebusiness application, browses its contents, orders a product, and then leaves the application. User interaction happens when

users are interacting with their Web browsers,

but are not actually sending or receiving requests. Interaction time is the period from when the users receive a Web page to when they send their next request—the time it takes them to read and absorb the contents of the page, including the time it takes them to decide what to do next.

Setting up the Web load test with a load script will help in leveraging the user and system for exercising the system.

Stress Test Environment

As you set up your testing environment for a stress test, you need to make sure you can answer the following questions:

- Will my test be able to support all the users and still maintain performance?
- Will my test be able to simulate the number of transactions that pass through in a matter of hours?
- Will my test be able to uncover whether the system will break?
- Will my server crash if the load continues over and over?

The test should be set up so that you can simulate the load; for example:

- If you have a remote Web site, you should be able to monitor up to four Web sites or URLs.
- There should be a way to monitor the load intervals.
- The load test should be able to simulate the SSL (Secure Server).
- The test should be able to simulate when a user submits the Form Data (GET method).
- The test should be set up to simulate and authenticate the keyword verification.
- The test should be able to simulate up to six email or pager mail addresses and an alert should occur when there is a failure.

It is important to remember when stressing your Web site to give a certain number of users a page to stress test and give them a certain amount of time in which to run the test. This test will then go out to the server and should simulate a large number of users. A stress-testing tool can monitor this test, and a graph of the data can be used to monitor the

activity. The key here is to continue to increase the stress level by increasing the number of users until the system performance begins to decrease.

Some of the key data features that can help you measure this type of stress test, determine the load, and uncover bottlenecks in the system are:

- Amount of memory available and used
- The processor time used
- The number of requests per second
- The amount of time it takes ASP pages to be set up
- Server timing errors

Peak Load and Testing Parameters

Determining your peak load is important before beginning the assessment of the Web test. It may mean more than just using user requests per second to stress the system. There should be a combination of determinants such as requests per second, processor time, and memory usage. There is also the consideration of the type of information that is on your Web page from graphics and code processing, such as scripts, to ASP pages. Then it is important to determine what is fast and what is slow for your system. The type of connection can be a critical component here, such as T1 or T3 versus a modem hookup. After you have selected your threshold, you can stress your system to additional limits.

As a tester you need to set up test parameters to make sure you can log the number of users coming into and leaving the test. This should be started in a small way and steadily increased. The test should also begin by selecting a test page that may not have a large amount of graphics and steadily increasing the complexity of the test by increasing the number of graphics and image requests. Keep in mind that images will take up additional bandwidth and resources on the server but do not really have a large impact on the server's processor.

Another important item to remember is that you need to account for the length of time the user will spend surfing each page. As you test, you should set up a log to determine the approximate time spent on each page, whether it is 25 or 30 seconds. It may be recorded that each user spends at least 30 seconds on each page, and that will produce a heightened response for the server. As the request is queued, this will be analyzed as the test continues.

The Mock Test

It is a good idea to set up a mock test before you begin your actual test. This is a way to measure the server's stressed performance. As you progress with your stress testing, you can set up a measurement of metrics to determine the efficiency of the test.

After the initial test, you can determine the breaking point for the server. It may be a processor problem or even a memory problem. You need to be able to check your log to

determine the average amount of time that it takes your processor to perform the test. Running graphics or even ASP pages can cause processor problems and a limitation every time you run your stress test.

Memory tends to be a problem with the stress test. This may be due to a memory leak or lack of memory. You need to log and monitor the amount of disk capacity during the stress test. As mentioned earlier, the bandwidth can account for the slow down of the processing of the Web site speed. If the test hangs and there is a large waiting period, your processor usage is too low to handle the amount of stress on the system.

Simulate Resources

It is important to be able to run the system in a high-stress format so that you can actually simulate the resources and understand how to handle a specific load. For example, a bank transaction processing system may be designed to process up to 150 transactions per second, whereas an operating system may be designed to handle up to 200 separate terminals. The different tests need to be designed to ensure that the system can process the expected load. This type of testing usually involves planning a series of tests where the load is gradually increased to reflect the expected usage pattern. The stress tests can steadily increase the load on the system beyond the maximum design load until the system fails.

This type of testing has a dual function of testing the system for failure and looking for a combination of events that occur when a load is placed on the server. Stress testing can then determine if overloading the system results in loss of data or user service to the customers. The use of stress testing is particularly relevant to an ecommerce system with Web databases.

Increase Capacity Testing

When you begin your stress testing, you will want to increase your capacity testing to make sure you are able to handle the increased load of data such as ASP pages and graphics. When you test the ASP pages, you may want to create a page similar to the original page that will simulate the same items on the ASP page and have it send the information to a test bed with a process that completes just a small data output. By doing this, you will have your processor still stressing the system but not taking up the bandwidth by sending the HTML code along the full path. This will not stress the entire code but will give you a basis from which to work. Dividing the requests per second by the total number of users or threads will determine the number of transactions per second. It will tell you at what point the server will start becoming less efficient at handling the load. Let's look at an example. Let's say your test with 50 users shows your server can handle 5 requests per second, with 100 users it is 10 requests per second, with 200 users it is 15 requests per second, and eventually with 300 users it is 20 requests per second. Your requests per second are continually climbing, so it seems that you are obtaining steadily improving performance. Let's look at the ratios:

05/50 = 0.1

10/100 = 0.1

15/200 = 0.075

20/300= 0.073

From this example you can see that the performance of the server is becoming less and less efficient as the load grows. This in itself is not necessarily bad (as long as your pages are still returning within your target time frame). However, it can be a useful indicator during your optimization process and does give you some indication of how much leeway you have to handle unexpected peaks.

Optimizing the Web pages is another matter all together and may need to be handled by the programmers responsible for Web coding. To optimize the page, the programmer should make sure that the design is functional and graphically pleasing, information is correct and easy to obtain, confirmation is provided to buyers, and orders are tracked.

How Much Stress?

As you set up your Web load test, you will probably wonder to what extent you will need to stress your server. This tends to be unclear. It may depend on the server's line speed, the clients' line speed, and the average size of the document being delivered. The data rate that the server can provide and the server's capacity in terms of the number of transactions per second can be measured. Both of these measurements depend only on the server's performance. Keep in mind the line speed is not part of the formula that will measure the speed (modem speed can differ from T1 access). When the server's capacity is measured by the metrics, the server's ability can be determined.

When the server is stressed, it may not be able to handle transactions as quickly, or if the server is serving many clients, this will slow the speed of the lines. The number of transactions may rise for this type of data delivery. It is at a certain point of delivery that the transactions become an important issue. When the scheduling overhead percentage rises, the system stops rising and begins to decline in response to increased loading. A well-designed system will limit the number of transactions running together to prevent overloading, and of course it will be designed to complete transactions as quickly as possible. An important aspect of the load test is to be able to measure the delivery time for the transaction and the number of transactions that can occur simultaneously.

Testing Tools

There are several types of testing tools that can simulate hundreds of users at server connection speeds. For a Web tool to be successful, it should be used during the design and setup development phases. The software should be set up and used to conform to the Web server setup as well.

Segue's SilkPerformer

SilkPerformer can be used to test heavy performance loads; it has been used for the following scenarios:

Real-world simulations. SilkPerformer accurately emulates the most realistic ebusiness

conditions by simulating a nearly infinite number of simultaneous users and traffic scenarios with a single script. It can also simulate multiple combinations of protocols and computing environments using a single recorder to capture and replay scripts.

End-to-end reliability. SilkPerformer lets you determine your site's scalability from the earliest stages of development right through final production.

Firewall support. SilkPerformer maintains firewall integrity while monitoring all application and database servers across any wide area network or Internet infrastructure.

Agent health control. To ensure valid test results, SilkPerformer continuously monitors the CPU utilization, memory requirements, and responsiveness of each agent.

Mercury's LoadRunner

This tool has its own unique load testing capabilities. LoadRunner provides an automated, realistic, and reliable load testing solution, as follows:

Optimizes application performance across any application infrastructure. LoadRunner tests an entire enterprise infrastructure including ebusiness, enterprise resource planning (ERP), customer relationship management (CRM), and custom applications by emulating thousands of users, enabling IT and Web groups to improve application performance.

Scalable load testing. LoadRunner is a scalable load testing tool and can emulate the activity of hundreds of thousands of users with minimal hardware.

Accelerates problem resolution in production environments. LoadRunner features integration with Mercury Interactive's Application Performance Management tools. The same scripts created during testing can be reused to monitor your application once it is deployed.

Open and extendable. LoadRunner features an open API, enabling users or third-party vendors to integrate LoadRunner into the unique environments.

Figures 9.3 through 9.5 illustrate how LoadRunner is valuable for SAP load testing.

The results of Service Advertising Protocol (SAP) load testing its Employee Self-Service (ESS) applications illustrate several important points about load testing. First, despite estimations developers may have about the performance of their applications, they can only ensure scalability with a rigorous load testing approach. Second, using such an approach can lead to immediate improvement in application scalability.

Perhaps the most interesting result of this load testing exercise is that SAP can fine-tune the ESS architecture by distributing functions across different machines. Running the Web server and ESS Applications components on the same machine could only support 200 users. Two such machines could support a total of 400 users. However, allocating the Web server to one machine and the ESS Applications components to another machine could support 1,200 users. Even accounting for the difference in machines, distributing components across two machines still supports more than twice as many users than simply replicating all of the

components on both machines.

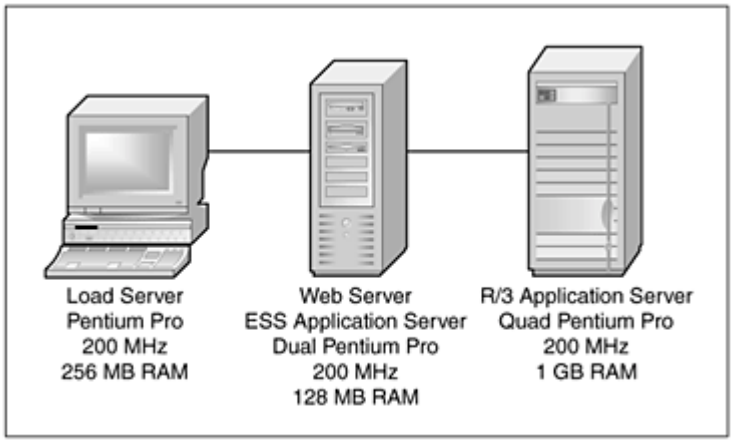

Figure 9.3 A low-level illustration of the architecture of a Web server, application server, and load server.

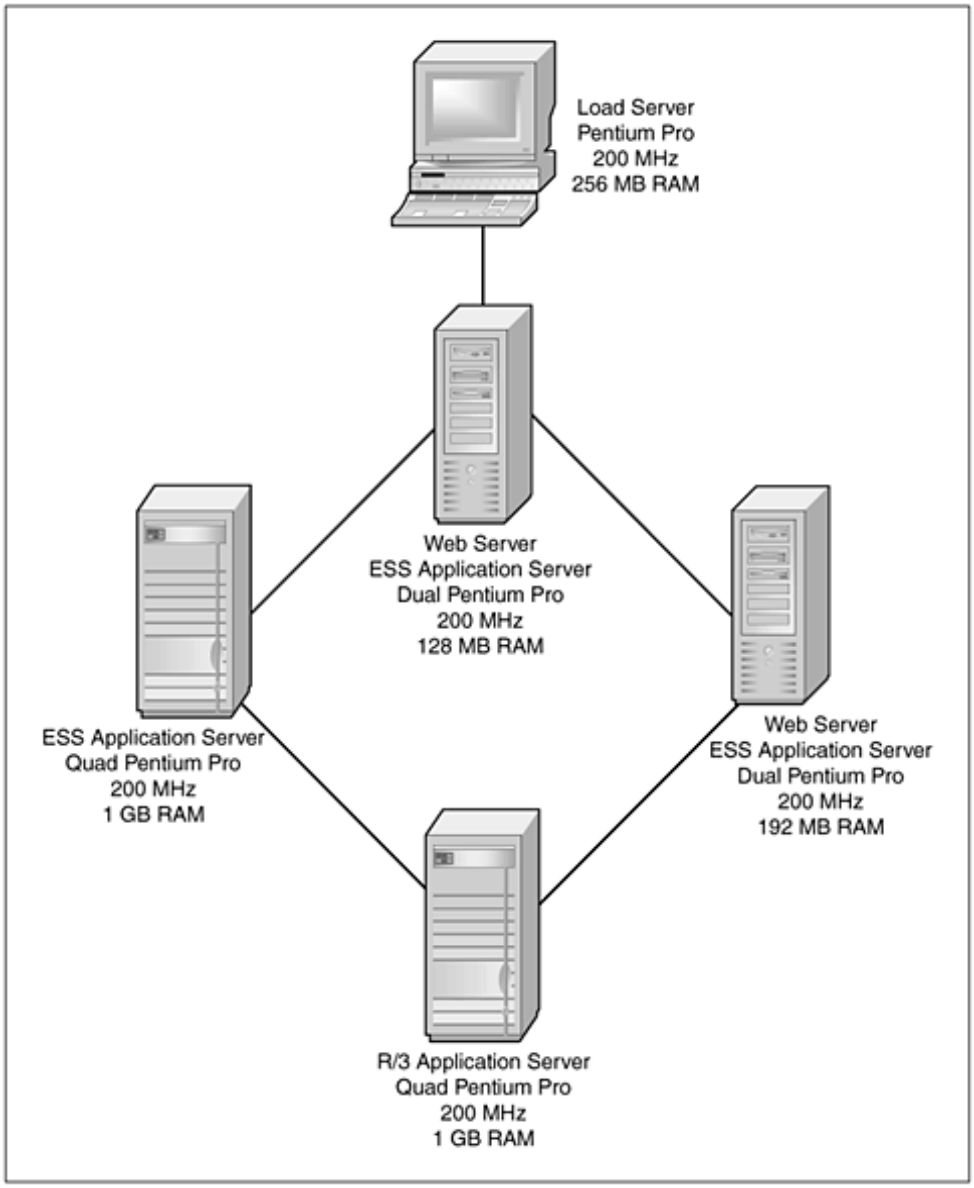

Figure 9.4 An illustration of a different architecture server setup.

The distribution advantage is important information for SAP's customers. Most importantly, it allows them to optimally allocate their server hardware to the correct software components. Moreover, partitioning discrete services across machines gives users the flexibility to add system resources where they can pro-

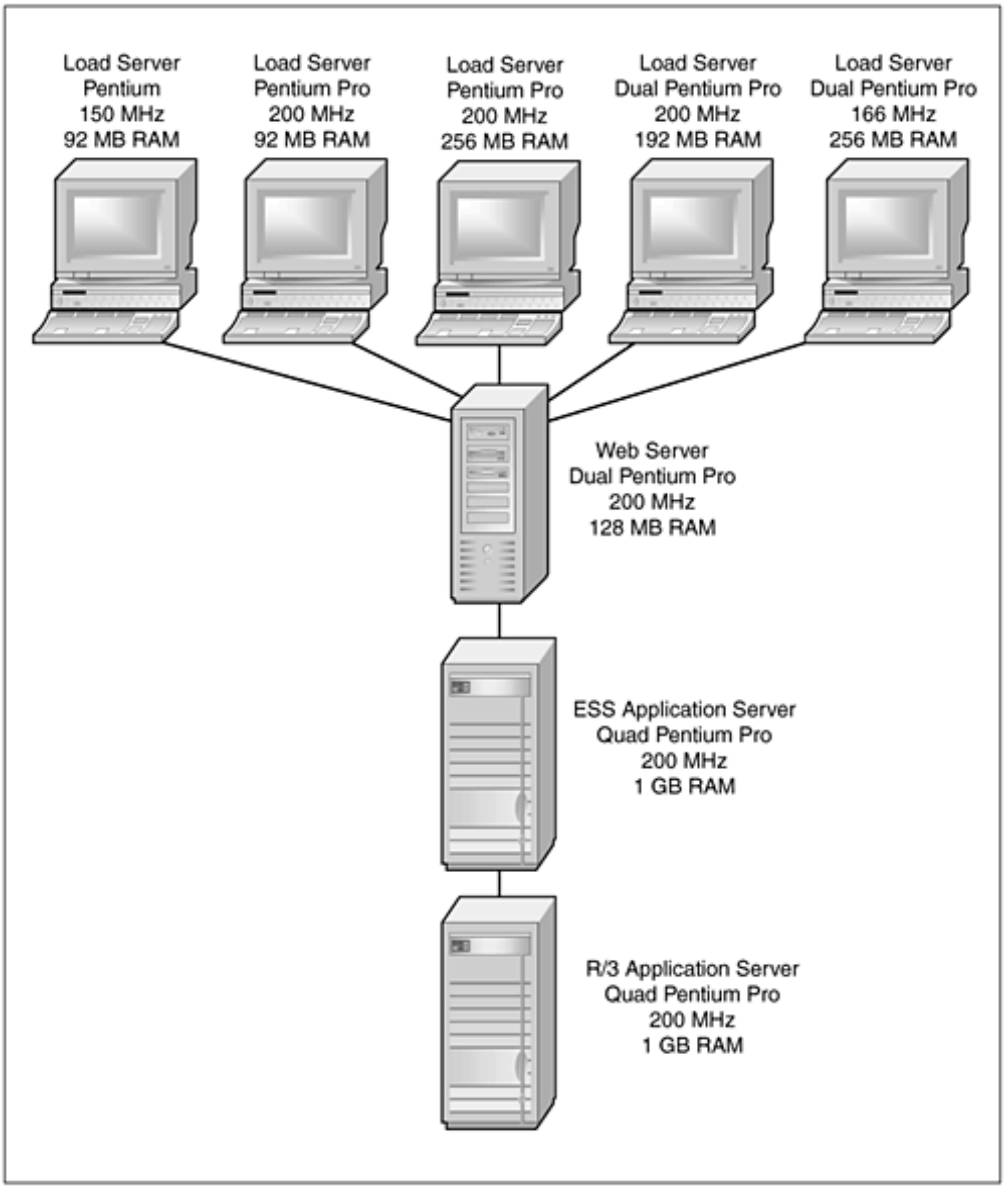

Figure 9.5 An illustration of a complex server setup.

vide the most benefit. The Web server will become the bottleneck first. At this point, users can add another Web server. If the ESS or R/3 Application servers become a bottleneck, users can add another one. This flexibility will allow ESS Applications users to scale their installations to tens of thousands of users. To

optimize such a large installation, these customers can use LoadRunner to further tune their architectures. LoadRunner helped SAP arrive at this extremely scalable solution with a very reasonable amount of effort. SAP achieved its impressive gains with two of its own programmers and one from HAHT Software in just a few weeks.

The results should give SAP customers a good deal of additional confidence in R/3's capabilities. Having 1,200 simultaneous users is substantial, even considering that organizations with tens of thousands of employees use R/3. Being able to support 1,200 users with only two high-end Pentium machines, one for the Web server and one for the ESS Applications components, is an admirable feat. Customers will incur minimal additional hardware cost when giving their employees Web access to R/3's human resources functionality. Costs will scale linearly with the number of employees using the application.

LoadRunner's SAP GUI interface support extends the possibility of such gains to the whole R/3 installation. As SAP's experience shows, such an effort does not require a host of programmers with specialized training or a great deal of time.

Microsoft Web Application Stress Tool

Microsoft's Web Application Stress Tool (WAS), which was developed by Web testers, is a simulation tool that is designed to realistically reproduce multiple browsers requesting pages from a Web application. The tool has been made as easy to use as possible by masking some of the complexities of Web server testing. This makes the tool desirable for anyone interested in gathering performance data on their Web site.

The Microsoft WAS tutorial (http://webtool.rte.microsoft.com) is one of the many different tools available for Web stress testing.

The Microsoft WAS Web stress tool is designed to realistically simulate multiple browsers requesting pages from a Web site. You can use this tool to gather performance and stability information about your Web application. This tool simulates a large number of requests with a relatively small number of client

machines. The goal is to create an environment that is as close to production as possible so that you can find and eliminate problems in the Web application prior to deployment.

Installing WAS

Copy SETUP.EXE to each client machine. Double-click on this executable to start the installation. Follow the setup wizard to complete the installation. Do not install WAS on your Web server because this may affect the performance of the server being tested.

Using the WAS Sample Script

After installing and opening WAS for the first time, a script called *Sample Script* is created. Use this script to get a feel for some of the features in the WAS Web stress tool. The actual files used in the Sample Script are stored in the folder where you installed WAS. Copy the Samples folder to the root directory of your Web site before continuing with this tutorial.

The left-hand window of WAS is known as the script view. This view displays all of the

scripts stored in the current WAS database. If this is a new database, the only items you see in the script view are *Defaults* and *Sample Script*.

There are seven script items in the Sample Script. Each script item uses a special feature of WAS. For example, notice that one of the script items is a POST. Using your mouse, highlight the row header to the left of the POST script item, and then double-click. This opens the script item details view. From this view you can edit the querystring name-value pairs, change POST data, modify the header, and enable secure socket layer. Figure 9.6 illustrates the Microsoft Web Application Stress Tool.

Take a look at the Querystring tab view. Notice that the names *User* and *Password* are being passed in the querystring. Also notice that %*Username*% and %*Password*% are the values being passed. These are not literal values; they are special variables that tell WAS to pass the next available user and next available password from the list. WAS automatically cycles through the usernames and passwords, passing the next set with each POST.

Close the script item details view by clicking the OK button.

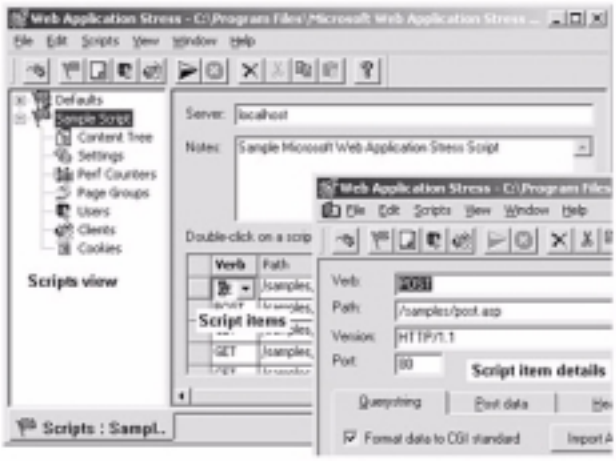

Figure 9.6 Microsoft Web Application Stress Tool.

Page Groups

The last two script items in the Sample Script contain the text "adGrp" under the *Group* column. This is a page group. A page group is shown as the default unless you change it. Page groups are used to reorganize the order in which the script items are invoked. It is also used to change the number of times that each script item is invoked while a script is running.

You can see a list of all the page groups by selecting the *Page Groups* node in the script tree view. You may also change the distribution percentages from this view. Notice that keep-alives are enabled for entire page groups at a time, which is illustrated in Figure 9.7.

Performance Counters

Select the *Perf Counters* node from the script tree and click on the *Add counter* button. It may take a moment or so to load the Add Counters dialog when this button is clicked for the first time. Add the following counters:

- Web Service: Get Requests/sec
- Web Service: Post Requests/sec

Page 217

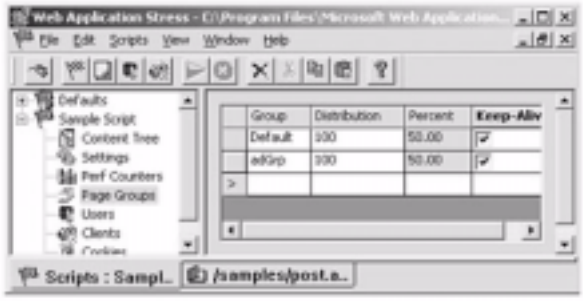

Figure 9.7 Page groups from the Microsoft Web Application Stress Tool.

- System: % Total processor time
- Active Server Pages: Requests/sec

Change the *Collection interval* to every 5 seconds. Capturing the correct performance monitor counters is critical to obtaining the correct data to analyze when the test completes. There are several important performance monitor counters to choose from (see Figure 9.8), based on the type of application you are testing.

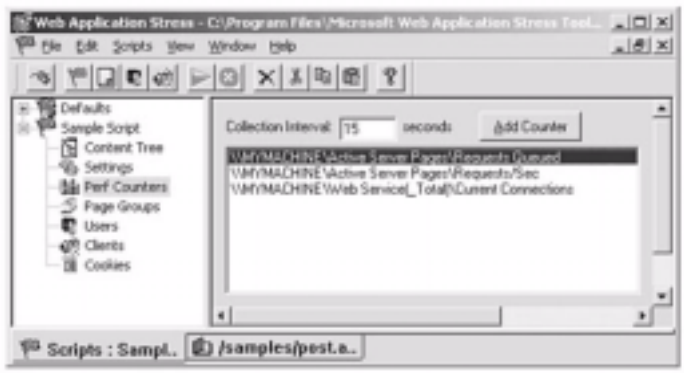

Figure 9.8 Collection intervals in the Microsoft Web Application Stress Tool.

Page 218

Settings

Select the *Settings* node and change the *Test Run Time* to 1 minute, down from 15. Leave the other settings as they are for the time being, but look over the other options in this view to get an idea of what can be configured for a script. You may also set the default settings (see Figure 9.9) for all new scripts by selecting the Defaults node and changing those options. Keep in mind that the settings in the Defaults node will not affect existing scripts, such as the Sample Script.

Users

Select the *Users* node and double-click on the Default user population, indicated by the icon shown in Figure 9.10. This opens the Users view where you can add and delete users from the default population and create new populations. Each WAS user stores cookie information and authentication data. Notice that there are 20 users in the default population. WAS users are not the same as *stress level (threads)*, a setting located in the Concurrent Connections section of the Settings node. The two concepts should be kept separate.

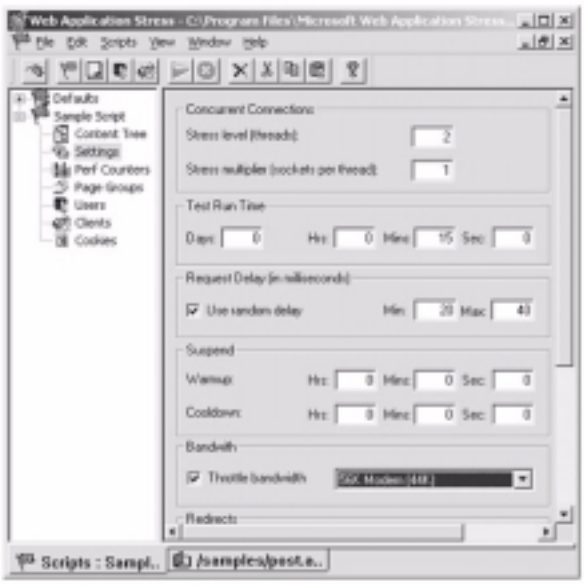

Figure 9.9 Setting nodes in the Microsoft Web Application Stress Tool.

Page 219

Figure 9.10 Icon of the Default user population.

Clients

Select *Scripts* from the *View* menu to return to the Script view. Select the *Clients* node under the Sample Script and double-click on the Default client group, indicated by the icon shown in Figure 9.11.

This opens the Clients view where you can add and delete client machines from the current group or add new groups of client machines. Notice that local-host is the only client and that it has a check box next to it. This means that the current machine is acting as a WAS client. Leave the Client view as is and select *Scripts* from the *View* menu to return to the Script view.

NOTE For more in-depth discussion of users and clients in Microsoft WAS, see the knowledge-base article, *Understanding Threads, Users, and Clients,* Web Application Stress Tool KB Article Number 32, http://webtool.rte.microsoft.com/kb/hkb32.htm.

Running a Test

Once you have created a script and configured all the settings, users, and clients, it is time to start the test. Select the Sample Script and choose *Run* from the *Scripts* menu. Allow the test

to complete.

Figure 9.11 Icon of the Default client group.

Page 220

Reporting

Select *Reports* from the *View* menu to open the Reports view. Expand the Sample Script report to display all of the report nodes. There should be at least one node whose title is the date and time at which your latest test was started. Expand and select the top level of this report node (see Figure 9.12), to view a summary of this test.

You can select a specific node of the report tree to view more in-depth information. For example, select the *Result Codes* node to display a sum of the HTTP result codes for every request in the test. Expanding the *Perf Counters* node displays the counters that have been collected during the test.

Expand the *Page data* node and select the first script item. The right-hand pane displays detailed information regarding this script item. Reports provide the response time for specific pages by determining when the script has finished downloading on the client. This is a good source of performance data.

The Time to first byte (TTFB) calculates the time in milliseconds from the request for the page until WAS receives the first byte of data. The Time to last byte (TTLB) calculates the total time in milliseconds from the request until the last byte of data has been received on the client. This number includes the TTFB time and any additional time needed to receive the last byte of data. All of the requests are sorted, and then the data is divided into percentiles.

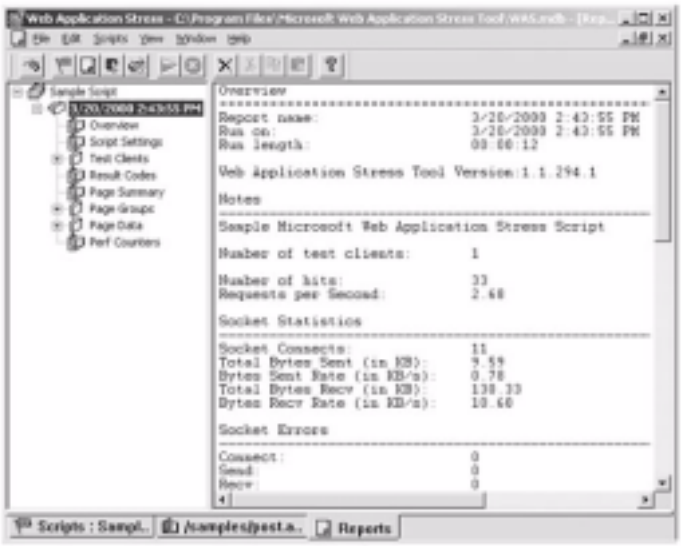

Figure 9.12 Report node.

Page 221

NOTE For more in-depth discussion of the p-squared algorithm and percentiles, see the WAS knowledge-base article, *Percentiles in WAS,* Web Application Stress Tool KB

Article Number 30, http://webtool.rte.microsoft.com.

While in Report view, you can select *Export to CSV* from the *File* menu. Exporting the report values to a format that Microsoft Excel can read allows you to create charts that show where most of the requests fall. For example, Figure 9.13 illustrates TTLB.

General Guidelines for Using WAS

The following are general guidelines for using WAS:

As a rule, use between 10 and 100 threads. You will rarely have a need to run a stress test that requires more than 100 threads per client machine. Be sure to monitor the processor utilization on the clients. Anything below 80 percent should be okay. The client machine(s) may not be capable of sustaining the stress loads where the processor utilization is greater than 80 percent, at which point the test will become invalid.

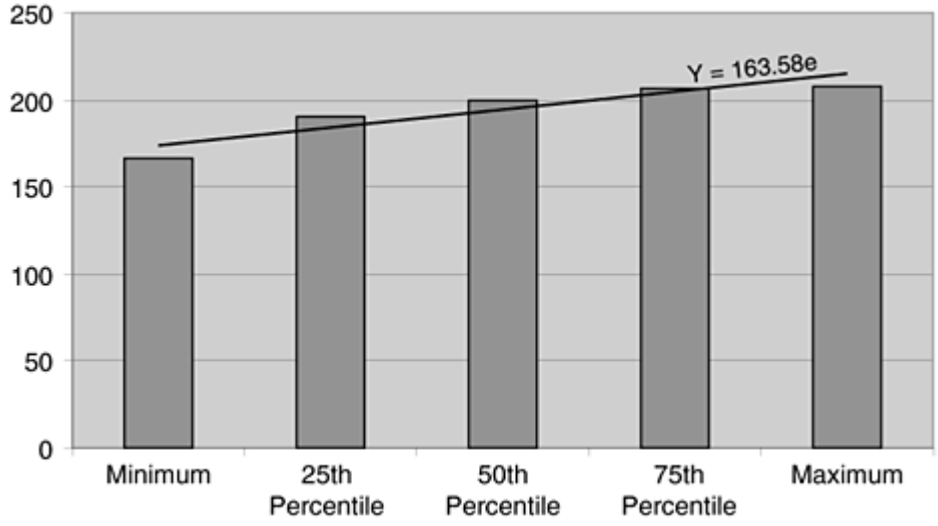

Figure 9.13 Time to last byte (TTLB) for DEFAULT.ASP.

When adjusting the threads and sockets for a WAS stress test, use just one socket (stress multiplier) unless you are performing a special type of test. See the online help topic "Stress level vs. stress multiplier" if you require more information on this setting.

Limit the number of users to less than 1,000 unless there is a specific reason that you require more unique users. Although the number of users allowable is only limited by the amount of memory on the client machine, you may find that it takes too long for a test to initialize when using a large number of users.

Avoid creating scripts with more than 1,000 script items. The number of scripts is only limited by the amount of memory on the client machine, but you may find that it takes too long for a test to initialize when using a large number of script items.

Summary

This chapter shows how load and stress testing is a critical component for all Web testing. By performing Web load testing, you will be able to find performance bottlenecks in your design

and setup during the early stages of development, and you will be able to efficiently run your Web site. A load test tool can emulate hundreds or even thousands of users accessing a Web site during short periods of time. There are several tools available that emulate the load and stress that your site will encounter. It is important to take the time to evaluate what is available and find the tool that will enhance your Web site.

Chapter 10 will demonstrate the actual test process.

CHAPTER 10
Running the Web Test

A Web test is conducted to ensure that the Web site runs according to the design and requirements. To run the test properly, you should understand the basic testing process. The tester should be able to carry out and run the actual test with the tools and methods you have chosen. This chapter provides several steps that will help guide you through the Web test process.

Understanding the Life Cycle for the Web Application

When developing an application in any language, it is essential to understand the life cycle that is necessary for the creation of Web application. Following the life cycle will assure the customer that the Web application meets or exceeds the customer's expectations. You may ask, How is this important to testing? A part of the tester's job is to ensure that the Web application meets or exceeds customer

satisfaction, so a tester should understand the life cycle development phase. Following are four phases to the Web development life cycle:

- Planning phase
- Analysis and design phase
- Implementation and testing phase
- Installation and maintenance

It is best to work through all of these phases of the life cycle. This will allow the developer, tester, and manager an opportunity to voice concerns and provide input to the process as they work through the process.

How to Plan the Web Test Phase

The planning phase addresses the problems and desired solutions for the Web-based project. The results of this phase should assist in the development of the requirements document. The requirements are a list of statements that define the use of the application. There are five steps to this phase:

Develop a statement of the problem to be solved. What exactly do you want to achieve by developing and testing this application? Following are items that should be considered:

- Description of present scenario
- Present problem constraints
- List of goals to be achieved

Develop a computerized solution strategy. This involves mapping out a prototype solution to the application.

Lay out the hardware, software, and resources involved. This can be handled with a management tool such as Microsoft Project.

Determine system goals and the requirements for the development process. The system goals should be defined and the requirements should be established before the development process begins.

Establish a sign off so that all parties accept the solution and business requirements. The sign off process will assure that all parties involved are satisfied and the system is certifiable and complete. Once it is certified it will be ready for the audit team to review.

Planning the process for development is critical to the success of the project. During this phase a project management tracking tool such as Microsoft Project should be selected. With it you can track resources, critical path dates, beginning and ending dates, and time estimates for the project. The management tracking tool uses information entered into it to calculate cost and duration estimates for the project. The project manager needs to consider the following when developing the Web project plan and working through the planning process:

- Life cycle models
- Organizational structure and resources
- Cost estimate
- Testing requirements
- Documentation
- Installation and maintenance

Life Cycle Models

The life cycle model is the series of steps through which an application progresses to a completed state. The following are the different types of life cycle models that can be used. You can decide which of the following life cycle models best suits your situation:

Waterfall. This model (see Figure 10.1) is a set of overlapped phases that a project sequentially goes through. As the project progresses through a phase, a decision is made to determine whether the next phase can be started or whether there should be a continuation in the present phase.

Page 226

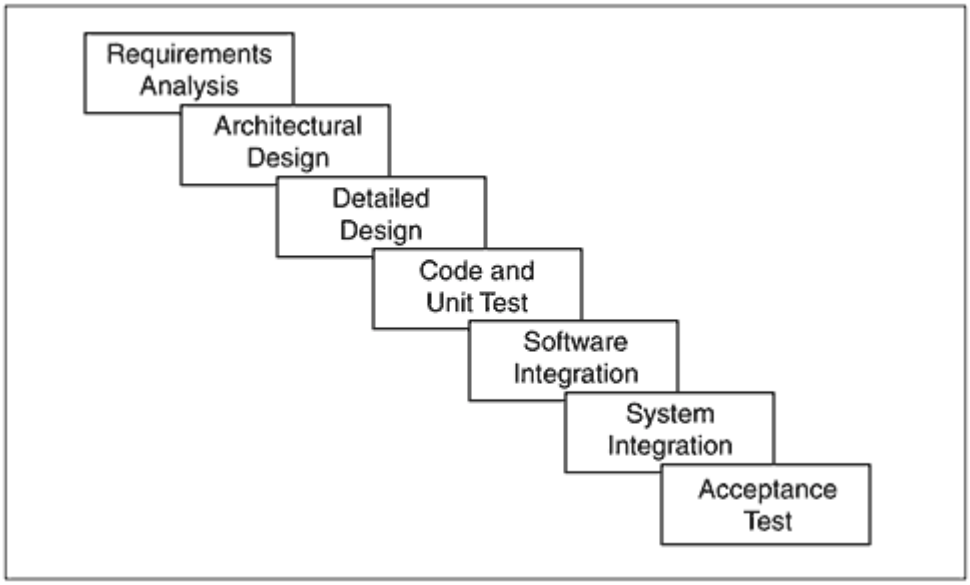

Figure 10.1 Waterfall model.
www.teleport.com/~qcs/papers/p821.htm

V-shaped. This model (see Figure 10.2) is similar to the waterfall method except that it stresses testing up front instead of in the later stages of development.

Prototyping. This model (see Figure 10.3) is used when a quick implementation is needed during the software requirements phase. This method provides feedback to the developers on strengths and weaknesses of a project so that a clear definition of direction can be taken.

Incremental. This model (see Figure 10.4) allows developers to construct the software in incremental stages, with each stage providing additional functionality. As you build your application, you verify and test through each phase.

Spiral. This model (see Figure 10.5) is used when risk analysis is considered in all iterations. Each spiral addresses major risks that can be identified in each phase.

Page 227

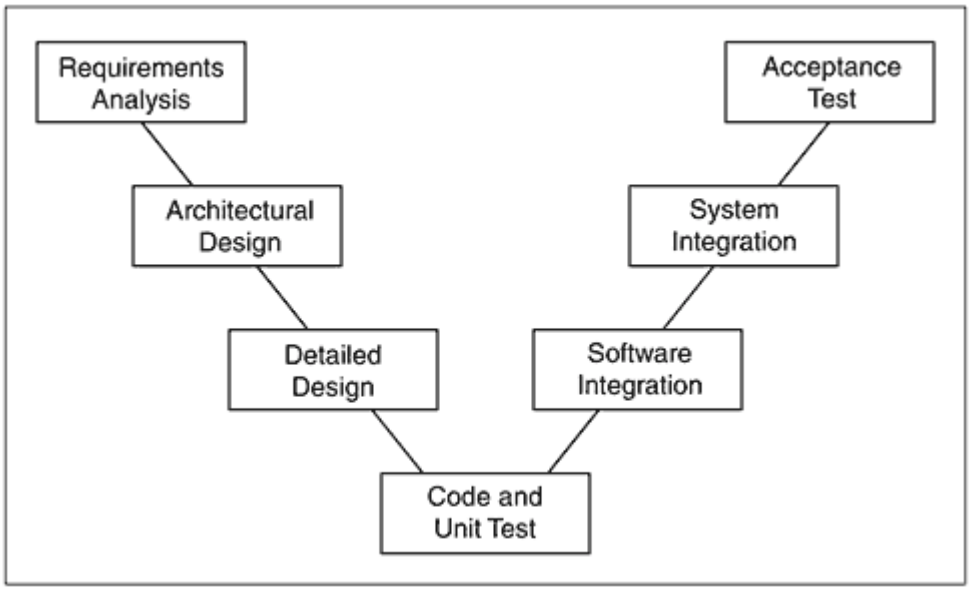

Figure 10.2 V-shaped model.
www.teleport.com/~qcs/papers/p821.htm

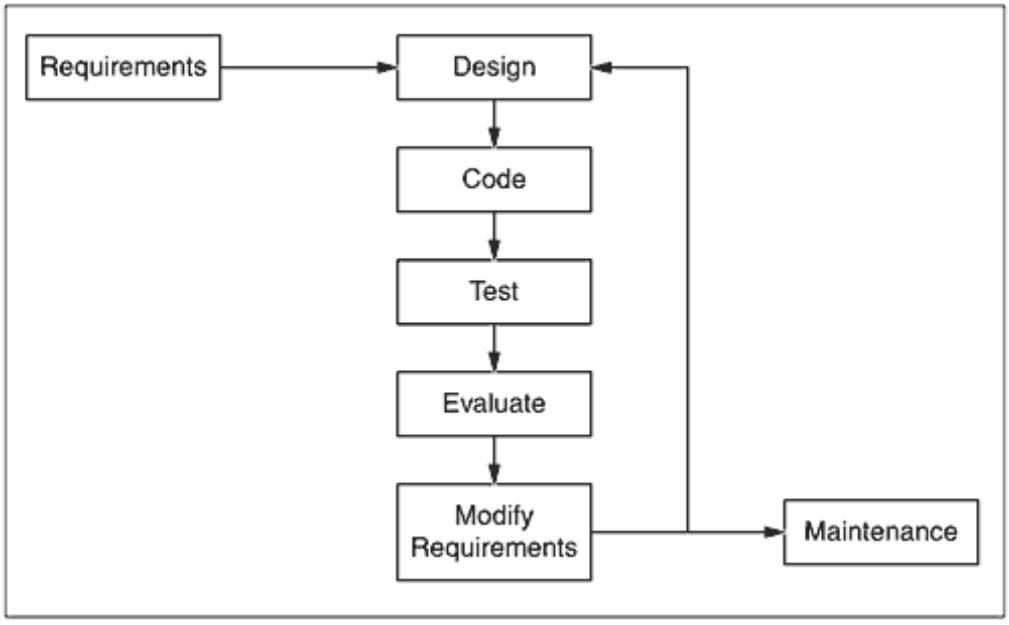

Figure 10.3 Prototyping model.

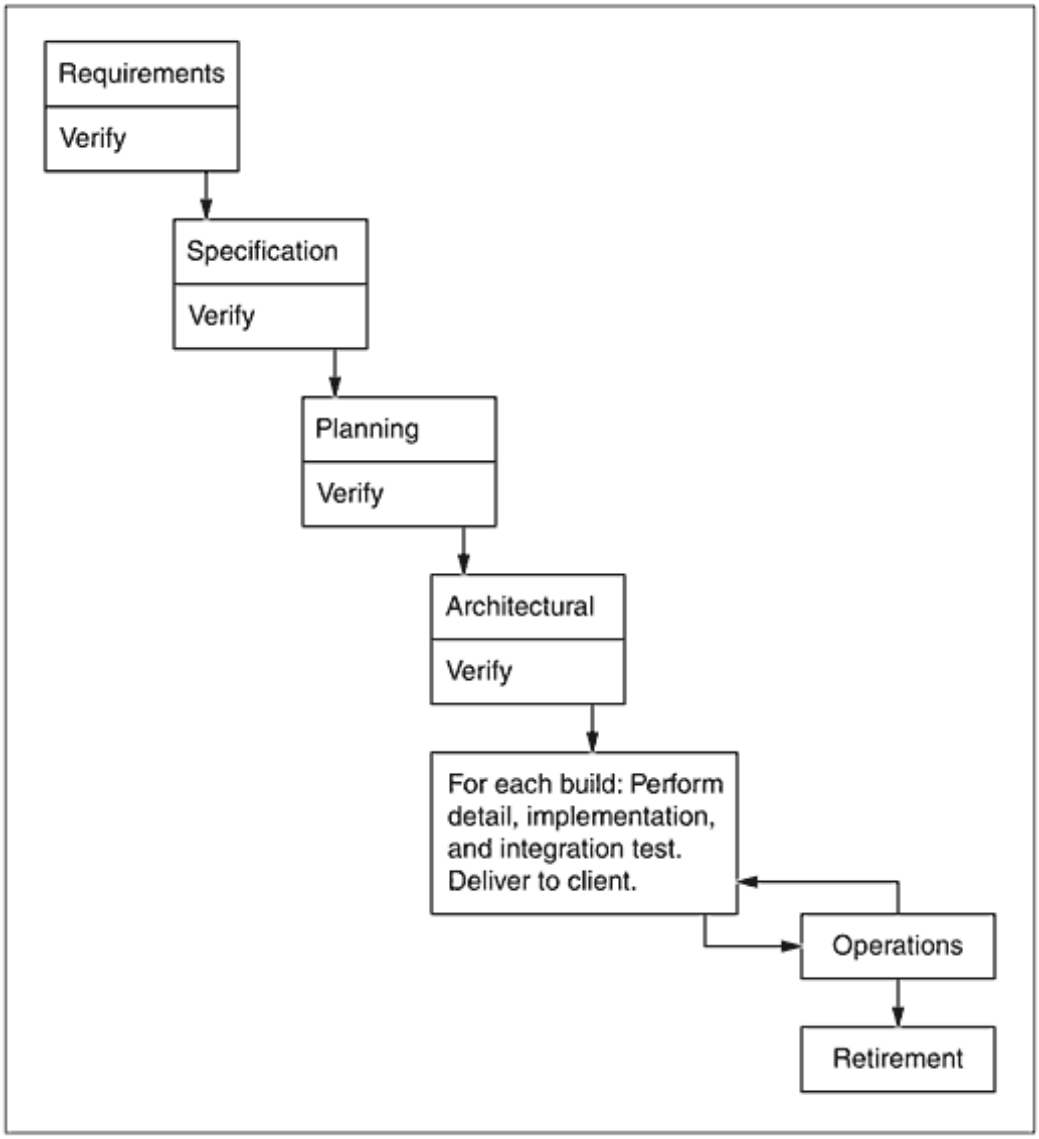

Figure 10.4 Incremental model.

Organizational Structure and Resources

The organizational structure needs to be defined so that involved parties know their role in the process. The following structures and tasks should be defined:

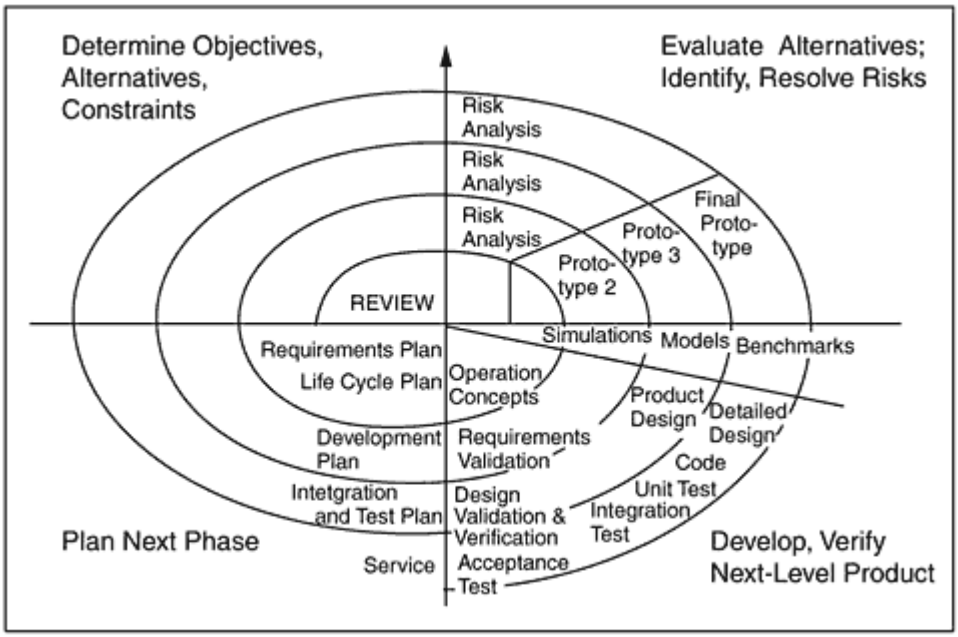

Figure 10.5 Spiral model.
www.cstp.umkc.edu/personal/cjweber/spiral/html

Management structure. Who makes up the management team and who is responsible for sign offs and acceptance of the different phases and plans?

Team structure. Who is a part of the team and what are the assigned roles?

Workload structure. Is the workload set up equitably and are all parties trained to fulfill their tasks?

Definitive work tasks. Are the work tasks all-inclusive and are all the tasks covered?

Roles should be assigned to members of the team to allow for fluent coverage of all tasks. It may be necessary to incorporate a backup or contingency plan for additional resources in case they are needed.

How to Make Accurate Cost Estimates

A cost estimate for the project should be made by the project manager. Each phase of the development should be analyzed and costs assigned, including the costs to perform the process, prepare products for that phase, and verify that products are complete. When making an estimate, you should take into account as many anticipated risks as possible. To make an accurate estimation for project cost, you should understand the modifications that will need to be made as the project progresses. Modifications may include revisions, enhancements, maintenance, upgrades, increased resources, and lost time.

Review and Implement the Testing Requirements

Documentation assures that all aspects of the application are cited. This documentation will be particularly valuable for using with the training process, and it will become especially

useful as the project is completed. This documentation can be used to develop and provide training for the parties involved in using the application. To prepare quality documentation, much research needs to be done. In Chapter 11, "Analyzing the Test Process and Documentation," we will cover documentation in more detail.

Installation and Maintenance Required for the Web Test

As the Web application nears completion, preparing to install the application is the next step to consider. The installation process should include system requirements, adequate hardware, appropriate configuration, adequate resources, and Web accessibility. When installation has been successfully completed, a scheduled maintenance session should be set up and monitored. This phase of the life cycle development is the groundwork for the next phases and needs to be laid out so that the tester can perform the appropriate tests.

Analysis and Design of the Web Test

So far we have addressed the planning cycle of software development. Completion of the planning phase is essential for the progression to the next step in the life cycle, which is analysis and design.

The analysis phase of the Web life cycle should include your project plan and site requirements. For example, let's say you are designing an accounts payable Web application. The requirements should be the basis on which to begin assessment of the application. We can begin with the following steps to define our business requirements:

Statement of the problem. How will we track online the new and old accounts payables to make sure we are paying our invoices on time?

Describe how the application will function. The accounts payable Web application needs to be able to accumulate data and produce a report that will show the dates for invoices and invoice due dates.

Define a solution. After entry of the data, a field will be required to track all invoice dates and due dates. Reports should be developed that will track specific dates based on queries sent to the system.

When a payment has been made, the information must be secured and credited to the correct account.

Figure 10.6 illustrates how you would define a problem (in our example, tracking accounts payable) and provide a description of the application and how the application can handle the input and tracking process. A solution can be derived and incorporated into a design.

After a data flow diagram is created, a performance measurement should be made that shows how quickly the sequence of events can be performed. The tester should also account for the stress and load the application will encounter. Along with the requirements, specifications for the project should be addressed such as:

- What will the Web application do?
- What will the Web application not do?
- Is the structure of the Web application correct?
- Will inputs produce expected outputs?
- Are there other possible solutions?

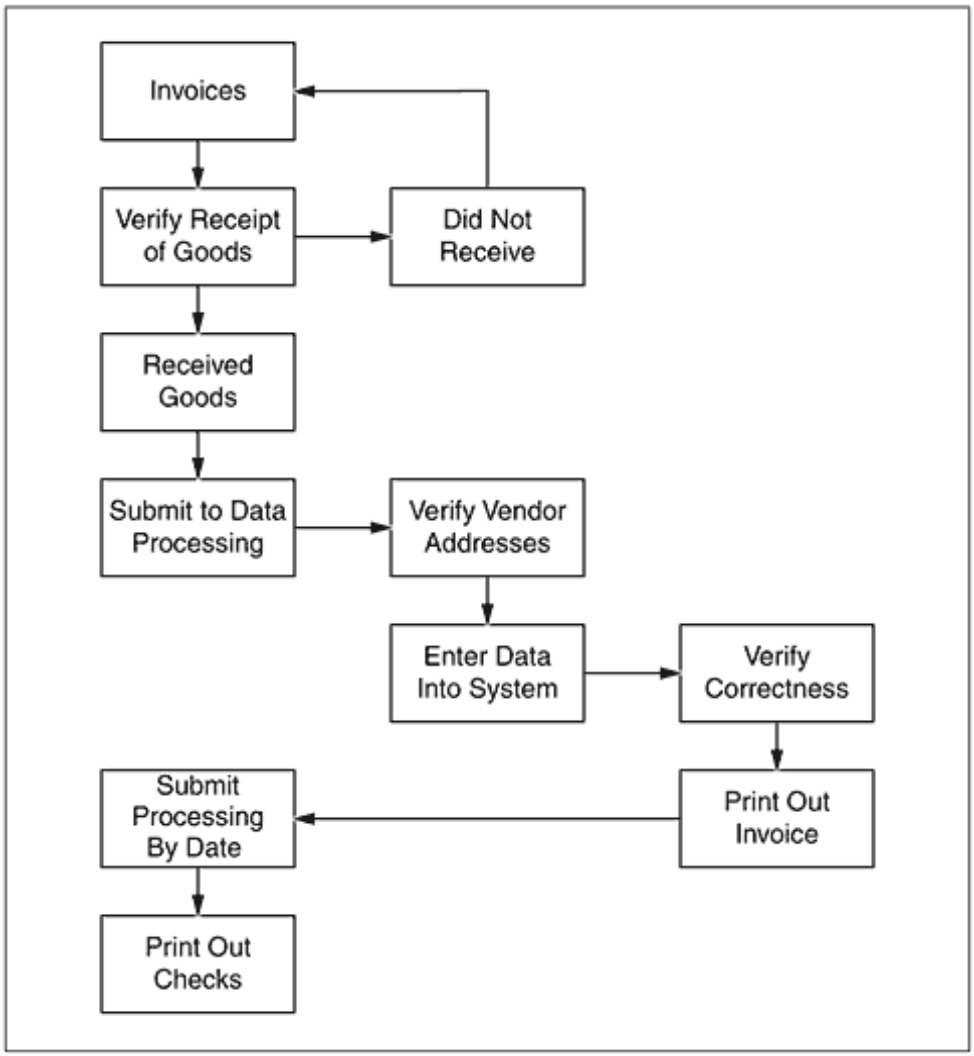

Figure 10.6 Tracking accounts payable.

Now that we have analyzed our problem and have created our data flow diagram, we can begin to consider how we will design the Web application. When we design the Web application, we should keep in mind that there are three different types of design activities (see Figure 10.7):

External design. External design involves the planning and specifications of the project, or characteristics of the project. This includes user displays,

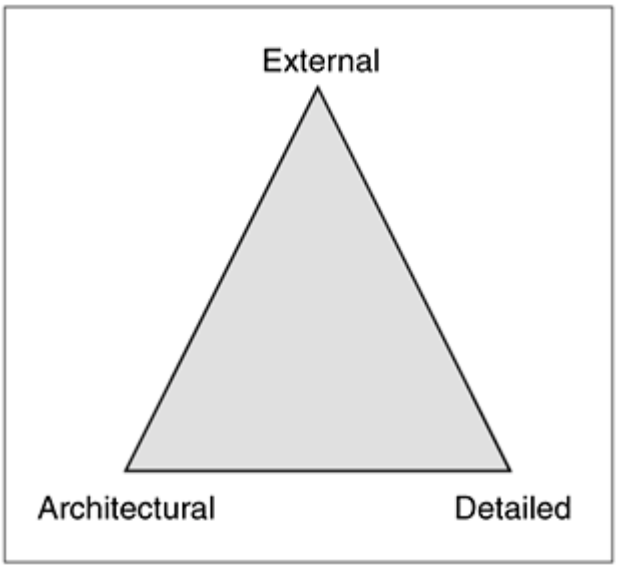

Figure 10.7 Design pyramid.

report formats, data sources, functional characteristics, performance requirements, and high-level structure of the project. External design begins with the analysis phase and continues through the design phase. External design actually involves planning out and specifying external structure, record design specifications, and test plans and provides a blueprint of implementation, testing, and maintenance.

Architectural design. Architectural design involves the conceptual view of the system, the internal processes, and high-level functionality and can be broken down into subfunctions. By using the test plan as your blueprint, you will be able to track the objectives of testing based on the functionality defined for external design. External and architectural design generally take a project from the requirements phase through to the preliminary design review.

Detail design. Detail design involves the specification and implementation of the functionality of the application. The schema for the application should be in place for the preparation of coding, which will begin with the implementation phase. The program at this phase should be broken down into foreseeable modules and procedures. At this phase documentation will become a critical component. The test plan should be reviewed and test cases should be designed in the detail design phase.

Implementing the Web Test

The implementation phase consists of taking the design specifications and translating them into code. During this phase, we would write source code in the language that was specified in the planning phase. We would also want to produce our internal documentation so that the code can be verified. Documentation takes on an important role during implementation and testing. Implementation can be seen as a series of steps, as illustrated in Figure 10.8. We will go into more detail on types of documentation in Chapter 11, "Analyzing the Test Process and

Documentation."

The implementation phase requires a set of business requirements, an architectural design, and a detailed design. Most of the problems that occur when implementing a software product are due to inadequate analysis and design. If

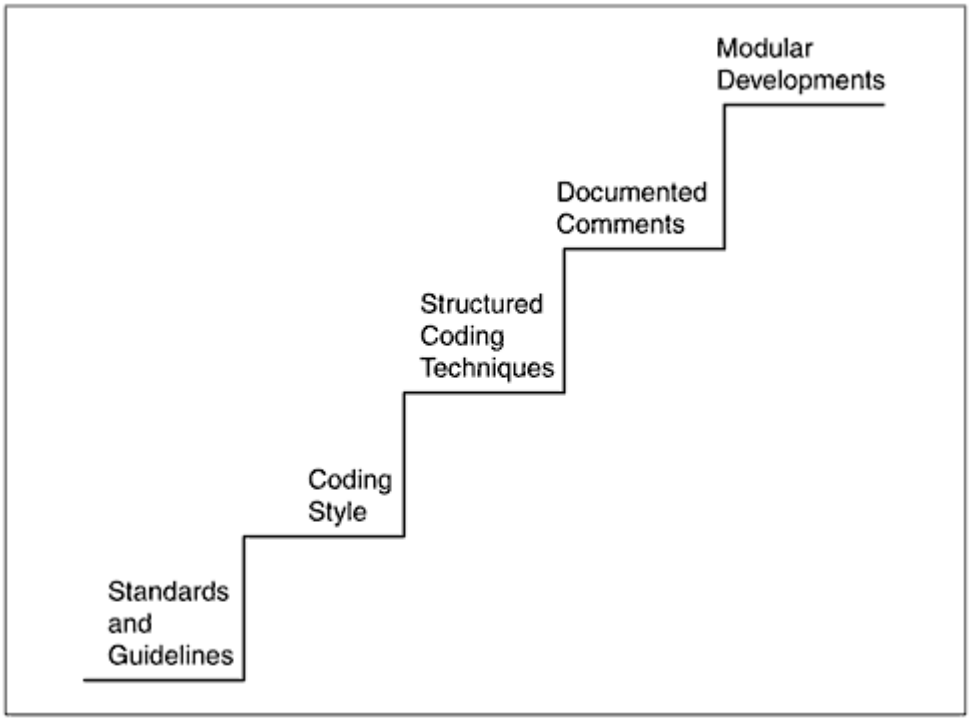

Figure 10.8 Steps in implementation.

a systematic approach is used, more reliable code will be produced, which will simplify the implementation process. This will also make debugging and testing of programs easier.

We covered the test plan in Chapter 1, "The Web Testing Process," but did not provide some important basic information about running the test. With the testing phase it is critical to have access to the business requirements; they are critical to and necessary for a successful outcome to the Web application life cycle. It is best to take the time to lay out business requirements before design and development of the Web application begins.

Business Requirements

For most applications, requirements need to be testable, that is, a tester should be able to take a business requirement and verify that it is correct according to one or more of the following techniques:

Demonstration. Execution of requirement where results can be reviewed.

Test. Execution after hardcopy results.

Analysis. Verification of results after execution.

Inspection. Review of code.

Along with the preceding techniques, business requirements should be:

Complete. Each requirement should stand alone.

Consistent. Each requirement should be similar with relation to format, content, and meaning and interrelated with the other requirements.

Correct. Each requirement should correctly translate from the original planning agreement.

Nonambiguous. Each requirement should avoid several interpretations.

Noncompound. Each requirement should avoid complexity; it should only have one meaning.

When you have a good set of requirements, it is time to decide how you will begin testing. This should be laid out in your test plan under test approach and strategy. The question for management and testers is: When is there enough testing? Realistically you will not be able to test every aspect of your application. You will have to decide which requirements you will want to develop into test cases for testing. There are a few guidelines that will help you make this decision:

Functional tests. This type of test will evaluate a specific operating condition using inputs and validating results. Functional tests are designed to test boundaries. A combination of correct and incorrect data should be used in this type of test.

Performance tests. This type of test should be designed to verify response and execution time. Bottlenecks in a system are generally found during this stage of testing.

Stress tests. This type of test is designed to identify possible overloads to the system such as too many users signed on to the system, too many terminals on the network, and network system too slow.

Structured tests. This type of test is designed to examine internal processing logic of the software system. It will go through the path of a selected routine to establish thorough testing.

Combinations of these types of tests are recommended when deciding on the type of testing you want to pursue.

Installation and Maintenance

In the installation and maintenance phase we will deploy our application and monitor changes that need to be made to the application. When the application is released and put into production, it should be monitored for the following:

Maintainability. Are changes easy to make and is code easy to read?

Reliability. Does it do unexpected things?

Efficiency. Is the design efficient?

Usability. Is it user friendly?

Portability. Is it easy to use on different systems?

Reusability. Can it be reused?

Maintenance

Maintenance activities occur after the product is in the hands of the customer. Throughout the maintenance phase, testing is still a critical need. As upgrades and changes are made, the application needs to be tested to ensure that it is still working properly and that the customer is satisfied with the product. Following are some important maintenance activities:

Enhancement to the product. Provide new functionality, improve user displays, and develop upgrades.

Adapting product to the environment. Move software to another system and modify to adapt to another environment.

Correcting problems. Modify and revalidate.

A maintenance schedule is often set up to provide the customer with assurance that support is available and problems will be handled. The maintenance portion of the life cycle consumes a large portion of the cost for software development because of the length of time involved. Generally, the maintenance stage can be anywhere from 2 to 10 years, depending upon the agreement made with the customer.

One of the most important components of the maintenance phase involves tracking and controlling activities. A maintenance schedule may consist of the following activities for implementing changes to the application. The change request is initiated, and then the request is analyzed, submitted for approval or denial, performed, tested, and implemented. Users should be able to initiate this process when they feel a correction needs to be made.

NOTE There are tools that can track and control versions of the product; they can be most useful during the maintenance phase. Chapter 5, "Web Site Testing Tools," and the CD-ROM list these tools and provide addresses for Web sites where you can find documentation and information that will help with this process. You can also find out about the types of training that is available to the tester and users.

Web Tester Skills

To carry out the Web testing process you must have an experienced tester and a project manager who will allow the tester to perform the tasks. The tester must be able to take a requirement and design a test plan according to the anticipated input and output of the Web site. The tester needs to understand and analyze the requirement and develop a methodology

that will cover all the issues involved. The tester should have a combination of some of the following experience:

- Certified or certifiable
- Certification from a Web vendor
- Regular attendance at Web testing seminars
- Researching ongoing changes in testing
- Relevant ebusiness experience
- Writing articles on testing
- Presenting papers to testing conferences
- Training in Web testing
- Degree in software engineering

Web Test Involvement

When you have a qualified tester and/or test team, you should involve them in the early stages of the testing process. While tracking down the answers to their questions, testers will inevitably find and report specification errors and code faults and maybe identify a few errors that can be eliminated before they create problems. This will provide the tester with time to plan and prepare tests, to identify and build or acquire the necessary tools, and to make sure that the testing is as thorough as possible.

Without early involvement, the product will be incompletely tested at best, and the schedule may be affected when testing does take place. This leads to poor performance by your Web site and additional costs, which reinforces that testing is an important part of setting up and implementing the Web site.

Degree of Testing

The tester must be able to analyze the situation and determine the necessary degree of the testing. Because the tester knows before the start of testing that there will be some faults in the code, it is possible to use tools such as memory analyzers to prevent or drastically reduce entire classes of errors from entering the functional test. The degree of testing will depend on how many faults are uncovered in the preliminary analysis.

A trained tester can track where the faults occur and attempt to identify and prevent the errors behind the faults. This will significantly increase the quality of the code. Fault analysis has made organizations improve the way they do business on more than one occasion. Analyzing faults will help to eliminate the errors and reduce the strain on your schedule and budget. Once you have determined the necessary degree of testing, you are ready to carry out

the test.

Carrying Out the Web Test Process

Both the designers and the testers should carry out the Web test process; it should be a collaborative effort. Although one or more teams should be charged with testing every aspect of the Web site that is delivered, testing for faults should also be a part of the responsibility of the designers. The designers and coders should be expected to perform low-level, code-based, and some of the regression testing. The designers and coders should then meet and work with testers to track bugs and problems.

The testers and developers should work together to perform integrated system testing of the completed system, including usability and maintainability, configuration, and reliability testing. During this development process there should be a parallel test process that checks the product and confirms that it is ready to move on to the next stage in development. The parallel testing will save the company money in the long run because most errors and issues will be resolved through this phase of testing.

Different tools are used throughout the testing process. Some are used to monitor and keep requirements, manage the process, and track bugs. The automated test tool is used to capture and house scripts for reuse. Keep in mind the automated tool is also used to help streamline work. If you find the tool is just used to maintain and house manual scripts, you are not using it efficiently.

Automating the test process is great, but it is important not to forget the primary goal of testing, which is to uncover errors in the Web site's functionality, usability, and performance. You may not want to automate the entire test process, and you may want to only use specific portions of the test tool. The automation of test scripts is best used for tests that involve repetitive scripting that may otherwise be monotonous for a tester.

Automation Process

The next two subsections discuss automation using different tools.

Cyrano

Following is specific information from the Cyrano Web site (www.cyrano.com) on the automation process using different test tools for performing the Web test process.

Cyrano creates, develops, and markets leading tools and solutions focused on the quality, performance, reliability, and change management of software applications. These applications may be running client-server (Windows PCs connected to Unix or Windows NT Servers), ecommerce (intranet/Internet), or mainframe (dumb terminals in character mode) environments.

Cyrano is one of the world leaders in the software testing, monitoring, and quality assurance marketplace. By providing industry-leading software for Web performance testing (OpenSTA), functional and regression testing, and application monitoring (WebTester), database performance testing and analysis (Impact and Workbench), database monitoring

(Production), security testing (e-secure), legacy understanding and testing (Wincap and Test), and strategic consulting and education offerings, Cyrano is an end-to-end quality assurance provider to its customers, helping them maximize their IT investments and ensure uninterrupted ebusiness. Cyrano offers integrated solutions, service, and support to companies that want to minimize risk, benchmark service-level agreements, and enable capacity planning for their IT infrastructures.

Cyrano also creates, develops, and markets mapping and documenting solutions for global IT resources to make information systems easier to access, deploy, manage, and maintain.

The Cyrano WebTester gives you different access methods and scenarios that need to be tested to ensure the reliability of your application. This will dramatically increase testing complexity and the amount of testing that must be done,

making automated testing the only viable alternative. The advantage of using a tool like Cyrano WebTester is that it has been specifically designed for the concepts of Web-based applications and is suited for all ebusiness systems. Its interface and concept of visual scripting mean that no technical training or programming is required, making it easier to use than any other solutions. In addition, the reusability of the testing scenarios throughout the life cycle of your application allows you to leverage your investment. Cyrano WebTester includes a 24 × 7 availability testing tool, to ensure that your application is continuously up and running, as expected.

SilkTest

SilkTest, from Segue Software, can be used for automated functional and regression testing. When you're testing a Web site, Java, or traditional client-server application, SilkTest will support most of your needs. SilkTest offers test planning and management, direct database access and validation, a flexible object-based fourth-generation 4Test language, a built-in recovery system for unattended testing, and the ability to test across multiple platforms, browsers, and technologies. You can use the power of test automation to accelerate your testing process and deliver a high-quality application to your customers.

You can also test your entire application end to end with SilkTest—from front-end clients to back-end Web, database, and application servers. You can drive your scripts from a central point of control, even when operating on entirely different platforms, which will give you an accurate picture of how well your system components are working together. SilkTest lets you choose recording or scripting to create your application tests and helps you plan tests, report on progress and, with SilkRadar, to track defects as well.

SilkTest's distributed testing architecture allows you to run a single test to traverse Windows and Unix clients, browsers, and Java-based systems such as network computers. With it you can verify workflow accuracy, perform concurrency testing, and ensure the accuracy of cross-platform transactions.

The component-based development allows for the mix-and-match integration of a diverse array of technologies. Test engineers have the ability to conduct in-depth testing across

heterogeneous components. SilkTest recognizes the multiple technologies that are found in ebusiness applications, including HTML, JavaScript, ActiveX, Java, Windows 98 controls, Visual Basic, and C + +. Because SilkTest is not tied to any specific vendor API, your scripts will continue to operate reliably as you adopt more technologies into your distributed applications.

Testing Java applets or components across multiple environments from a single script can be done with SilkTest. The tests can be integrated with the Java Developers Kit (JDK) and SWING user interface components, so you can test Java applications as they are being developed. The tests may directly interact with custom native Java tests to facilitate the testing of the wide variety of custom components in use today.

Web sites today are accessed from a broad variety of browsers, ranging from early non-table-based browsers to current Java-enabled versions. SilkTest allows you to validate all browser versions to assure accurate commerce transactions. Differences in browser suppliers, versions, and feature sets explode the number of configurations that need to be tested. With SilkTest, a single test script can be used without modification, including for multiple versions of Netscape Navigator and Microsoft Internet Explorer, to help you cut your testing time in half. With the single, cross-browser script you create, you can reduce test development and execution time and quickly test new software builds. Using SilkTest's Link Tester function, you can quickly obtain a listing of Web pages and links that make up your Web site. Simply enter the URL of the first page you want to scan and let SilkTest search through your pages for links and associated Web pages. You can then review various reports, errors, and actual Web pages. No set up or training is needed to use this feature.

Summary

As you can see, automating the test process will save you time and money. The scripts are reusable, the tools will validate the testing for you, and tracking errors is easier.

Chapter 11 discusses the importance of documentation to testers and how the test process is tracked on paper.

CHAPTER 11
Analyzing the Test Process and Documentation

Documentation is a set of written information and instructions about what an application does. It provides the specification of the program, trains users, and is used to write business requirements. Documentation can be used to give developers a guideline for revisions and upgrades and it is used throughout the test process to allow everyone involved to see a blueprint of the ongoing application process. Documentation is a way to provide the programmer with a guideline of what the software and Web site should do.

Analysis of the Test Process

Choosing good coverage goals and documentation can increase testing productivity. The highest level of testing productivity will occur when you find the most failures with the least effort. That is why you should document and prioritize each level and step of the test process. This effort is measured by the time required to create test cases, add them to your test suite, and run them. You should use a validation and verification coverage analysis strategy that

increases coverage as fast as possible. To accomplish this you should have an ongoing analysis of the test process represented by the different degrees of documentation necessary to carry out the process in a quick and orderly way.

Validation and Verification

Testers are continually trying to locate techniques to perform validation and verification (commonly known as V&V) to ensure that software will meet requirements and is free of technical errors. Can you provide validity and completely verify a Web site? Typically, the answer is No for verification and usually No for validity. A tester must be realistic about what can be verified and validated.

A verification technique is used to check every input/output possibility to ensure that a program operates correctly. The computer can read the code and follow the input through, or it can actually run the code with experimental input. The reading method is called white, or clear, box testing, and running the code method is called black box testing (discussed in Chapter 2, "Test Methodology"). Using this black box testing as a technique, it is impossible to apply all possible input test cases and evaluate all outputs. The tester needs to select a subset of test cases to exercise the areas of the code likely to be wrong.

Validation is running the test or a mock test; it is the computer-based testing process. Validation testing is designed to expose and take care of the errors built into the system.

Actual testing is considered a blend of verification and validation. Error detection cannot be done properly unless verification and validation are done together. Documentation is key to validation and verification testing because it will track, record, and resolve errors in the Web site before it is placed out on the Internet.

NOTE Sometimes validation cannot be performed completely. If the business requirements are stated precisely, it may be possible to go through the requirements document line by line and validate that the program has met each requirement. However, this is difficult and time consuming.

Documentation

After all of the testing is completed and tests are verified, documentation should be analyzed. Testing documentation is prepared for the entire testing process.

Documentation can be difficult to create because developers may not be able to use terminology that will be clear to other individuals. A technical writer should take the developer's language and create documentation that nontechnical people can understand. This

is why it is important to have a technical writer on your testing and development team.

Documentation should be prepared in an orderly format and include document outlines, tables, and figures to outline the specific descriptions of the application. Diagrams are essential to portray the specifics of the application. As the testing process progresses, the documentation can be used as a guideline for specific areas of the application. It is important to provide test document templates to use throughout the process to maintain a standard.

As testers prepare their tests, the documentation should be set up and used in a standard format. For example, IEEE standards and formats provide accurate documentation. Table 11.1 shows the types of documents that you should keep track of and the purpose of each one. It is important to keep track of all your documentation. You may want to make a file cabinet in a database to store all your documentation, or you may want to save all the documents as HTML files in a Web site so all team members can access the documentation through the Internet.

Test Plans

The objective of a test plan is to provide a road map so that the application can be evaluated through requirements or design statements for a specification design. A test plan is a document that describes objectives and the scope of a software project. When you prepare a test plan, you should think through the process of the software test. This document should be designed to help others understand the test application and process for testing. The document should be written so that it can successfully give the reader the full scope of the project. The plan should be thorough enough to be useful. Refer to the sample test plan in Chapter 1, "The Web Testing Process," for more information.

Table 11.1 Test Documents

DOCUMENT	PURPOSE
Test plan	Define the testing approach and resources and schedule the testing activities
Business requirements	Specify requirements of testing and identify the specific features to be tested by design
Test case	Define a test case identified by a test-design specification
Log for tracking test cases	Track test cases and test results
Test matrix	Track test cases and errors
Bug tracking report	Track errors as they occur and how they were corrected

Weekly status report	Give management a weekly progress report of the testing activity
Test script	Set up the sequential step-by-step test, giving expected and actual results
Issues log	Itemize and track specific testing issues and resolutions

Documents

Because you will be responsible for creating several types of documentation through the test process, the following sections provide some examples of different documentation templates. You will need to create templates that will be most applicable to your project.

Test Case Information

Figure 11.1 illustrates sample test case information. It provides the tester with the function that needs to be tested. For our sample we will be testing the functionality of formatting text. The form gives the tester the test case ID. This ID should be used for all documentation and correspondence concerning this test. The document then tells the tester what the actual test will consist of, what environment the test will be performed in, and the steps for the execution of the test and the expected outcome.

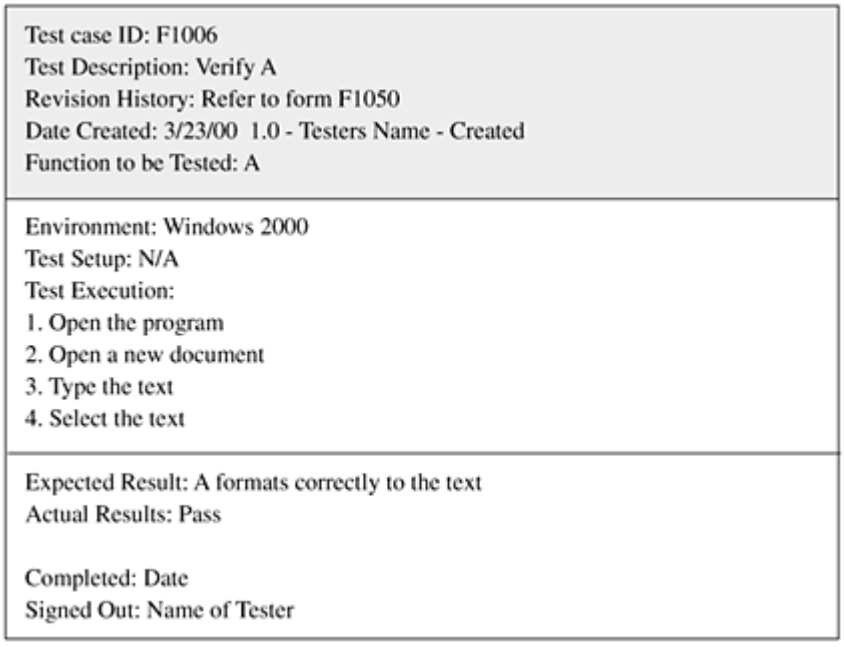

Figure 11.1 Sample test case information.

Test Case Form

The test case form (see Figure 11.2) is used to track all the test cases. It should include the test case numbers for all the tests being performed, the name of the test case, the process, the

business application condition that was tested, all associated scenarios, and the priority of the test case. The form should also include the date, the page number of the particular form you are using, and the system and integration information. This form is important because it tracks all the test cases, allows the test lead or another tester to reference the test case, and shows all the pertinent information of the case at a glance. This information should be placed in a database or Web site so all members of the team can review the information.

Log for Tracking Test Cases

The purpose of the test log is to provide a sequential record of test activity. The log can track the date, function, test case number, list the test scenario, show desired results, and then show the actual results of the test. This log is

Page 248

Test Case					Page:
System:					Date:
Test Case #	Test Case Name	Process	Application Conditions	Associated Tasks	Priority

Figure 11.2 Sample test case form.

important because it shows the date of the test and what went in during the test process. Other testers can look at this log and gather information about future tests that will encompass the similar actions. This log should be available to all test team members. Figure 11.3 shows an example of a chart used for test case tracking.

Date	Function	Test Case #	Test Scenario	Desired Results	Actual Results

Figure 11.3 Sample test case tracking.

Page 249

Test Log Administrator

The test log (see Figure 11.4) tracks all information from the previous example. This log will track test log IDs, test case or test script ID, the test event results, the action that was taken, and the date the action was taken. The log will also document the system on which the test was run and the page number for the log. This log is important for tracking all the test logs

and showing at a glance the event that resulted and the action taken from the test. This information is critical for tracking and registering all log entries.

Test Matrix

The test matrix (see Table 11.2) is an important part of your documentation. It will track the test case, test description, test cases and samples, whether the test passed or failed, the number of bugs, the number used to identify the bug, and additional comments. The test matrix is key for remediation. The tester will pull up the test matrix to find out which test cases did not pass and can see at a glance what bugs were recorded. The test lead or manager also uses the matrix to see if the site is ready for implementation.

Test Log				
System:				Page:
Test Log ID	Test Case/ Test Script ID	Test Event Result	Action	Action Date

Figure 11.4 Sample test log.

Tabl

11.2

Matr

TES CAS FIL OPE #	TEST DESCRIPTION	TEST CASI SAM	PASS FAIL	NO. OF BUC	BUC NO.	COM
1.1	Test file types supported by the program.	1.1	P/F	#	#	
1.2	Verify the different ways to open file (mouse, keyboard, and accelerated keys).	1.2	P/F	#	#	
1.3	Verify the files that can be opened from the local drives as well as network.	1.3	P/F	#	#	

Bug Tracking Report

The bug tracking report (see Figure 11.5) is an essential piece of documentation for all members of the test team. It lets you see the existing bugs and what measures have been taken to correct those bugs. The report also allows you to see if you have a bug that was carried over from the previous version or if you have one that has not yet been reported.

Weekly Status Report

The test manager relies heavily on the weekly status report (see Figure 11.6), which reports what was accomplished the previous week, what will be accomplished the following week, and what issues that are of concern to the test process will need to be addressed. Each person should keep a status report, and the test manager should track the reports.

Bug Tracking Report
[Bug's Report Title]
[Steps Involved to Reproduce the Error]
[Expected Result]
[Actual Result]
[Note]

Figure 11.5 Bug tracking report.

Status for <Person's Name> Week Ending <End of Week Date>
This Week: 1. Details of the progress of the week, what was scheduled to occur and what was scheduled to occur that did not happen.
Goals for Next Week: 1. Details of what should be accomplished for the next week.
Issues: 1. Issues that need to be addressed and handled.

Figure 11.6 Weekly status report.

Test Script						
Test Script Number:					Priority:	
System Tested:					Page:	
Test Case Number:			Date:		Tester:	
✓	Step	Action	Data Entry	Expected Results	Actual Results	Test Log ID

Figure 11.7 Test script.

Test Script

The test script (see Figure 11.7) is used to describe what step-by-step action will be tested. In the automated process many tools will take a script and use it over and over for different phases of the test. When you manually test, you have to write a script for each test, and then each result must be entered manually into an automated scripting tool. The test script is the key to good problem and bug tracking and to a smooth test process.

Issues Log

The issues log (see Figure 11.8) can be used to make a reference to a document; it includes the type of reference, the priority of the particular item, and a description of the item. This document is used to identify and reference anything that is associated with the test process. It can also be used to document the priority of a particular item for the tester, who will know the importance of the reference.

Resolution

The resolution log (see Figure 11.9) is used to track issues and how they have been resolved. It uses the reference number assigned to previous documents, the status of the problem, the last action taken on the problem, and who took that action. It will also report who made the decision for the resolution and how

Ref #	Type of Issue	Priority	Description

Figure 11.8 Issues log.

the resolution will be handled. This document will show the testers what is being done for documented problems and if their test is contingent on the resolution of a previous bug or problem.

Bug Tracking Software

One way for testers to detect and track bugs is to use bug tracking software to track and detect errors. Some of the software allows users to turn in bugs to a Web site through the email, whereas others track and store the bugs in a database. Table 11.3 shows some of the products that can be used for bug tracking.

Ref #	Status	Last Action Date	Action	Parties Involved	Decision Made	Resolution

Figure 11.9 Resolution log.

Table 11.3 Track Softw

NAI	FEATURES	ADDITIONAL INFORMATION	WEB SITE
App Inno Man	This system will allow multiple development departments to keep a centralized database of all problems and developments that are encountered in product development.	There is no need to load software on every computer with Web-based technology.	www.i com
Bug	BugTracker is designed to allow clients to report bugs directly from your support Web site. All reported bugs are forwarded by email to the appropriate support manager.	This product has a built-in quality assurance (QA) ability that assures that problems are fixed and that the solution is verified.	www. bugtra com/ bugtra index.l

Defe	DefectTracker has a new layout and an email notification system that will revolutionize the industry.	This may be the product for you because it has no per-user costs and has a solid support team.	www. Defect com
Mer Inter	PVCS provides the people, process, and products that help your teams deliver quality software faster. PVCS integrates the three disciplines of software configuration management (SCM): version and build management, issue management, and process management, to make your entire development operation more competitive.	PVCS can perform scales from the project team to the enterprise and supports a wide range of platforms and development environment.	www.i com
Bug	BugCollector Pro 3.0 is a multiuser database specifically designed for keeping track of software bugs and feature	BugCollector Pro 3.0 includes many features, including a built-in report designer,	www.r com

Table 11.3 (*Continued*)

NAI	FEATURES	ADDITIONAL INFORMATION	WEB SITE
Bug (con	requests. With it, you can track bugs from first report through resolution and feature requests from initial contact through implementation.	customizable charts, user-definable filters, a multilevel project tree.	
Prob	ProblemTracker is a powerful, easy-to-use Web-based tool for defect tracking and change management. ProblemTracker delivers the benefits of automated bug tracking to any desktop in a familiar Web browser interface, at a price every organization can afford.	ProblemTracker may bring many different benefits to your organization.	www. netresultscorp. com/ fs_pbtrk_info. html
Clea	ClearQuest is a flexible defect tracking/change request management system for tracking and reporting on defects.	This tool allows different types of change requests throughout the development life cycle.	www.rational. com

Test and Test Web	TestTrack and TestTrack Web are bug tracking solutions to deploy within your enterprise.	TestTrack or TestTrack Web can be used to watch productivity.	www.seapine. com
PR Trac	PR Tracker helps manage software development projects by tracking software bugs, action items, and change requests with problem reports.	PR Tracker runs on Windows, Windows 95 and Windows NT.	www.prtracker. com
SWI	SWBTracker supports concurrent multiuser licensing at an extremely competitive	This tool allows remote submission of new defects via email or Web browser,	www.software withbrains. com/suntrack. htm

continues

Tab

11.3

Trac

Soft

(*Co*

NAI	FEATURES	ADDITIONAL INFORMATION	WE SITI
SWI (con	price, as well as many of the most important features developers and testers are looking for in today's bug tracking software: automatic email notifications with customizable message templates, complete issue life cycle tracking with automatic change history logging, custom report designer, and many built-in summary and detail reports.	customizable view filters, file attachments, tracking of estimated and actual hours for programmers and testers, customer-product information, and more.	

Elen	Elementool is an application service provider for Web-based software bug tracking and support management tools. Elementool provides its services to software companies and business Web sites all over the world.	Elementool's tools can be used on a daily basis by its customers, and are integrated in their product development process. No software is required.	wwv elem

Summary

As you can see, there is quite a bit of documentation available for Web testing. In this chapter, I have included samples of the forms that I use when testing. When testing, you may find that you want to customize forms to meet your needs. The key to effective Web testing is creating, organizing, and updating documentation.

PART Three Templates

In the pages that follow, you'll find a complete set of the templates discussed throughout the book, which you can use to jump-start the progression of your Web testing process. In addition, the companion CD-ROM includes customizable versions of the templates that you

can tailor to your business objectives.

This page intentionally left blank.

Test Script					
Test Script Number:					Priority:
System Tested:					Page:
Test Case Number:				Date:	Tester:
✓	Step	Action	Data Entry	Expected Results	Actual Results

This page intentionally left blank.

Test Bed		
Name of Application:	Date:	Lead Engineer
Dates Application Will Be Tested:	Anticipated Problems:	
Dates for Setting Up Test Bed:	Engineers Assigned to Project:	Additional Res
Software/Hardware	Version/Type	Problems

This page intentionally left blank.

Test Case
Test case ID: Test Description: Revision History: Date Created: Function to be Tested:
Environment: Test Setup: Test Execution: 1. 2. 3. 4. 5.
Expected Result: Actual Results:
Completed: Signed Out:

This page intentionally left blank.

Test Case				
System:				
Test Case #	Test Case Name	Process	Application Conditions	Associated Tas

This page intentionally left blank.

Test Case Tracking				
Date	Function	Test Case #	Test Scenario	Desired Results

This page intentionally left blank.

Test Log			
System:			
Test Log ID	Test Case/Test Script ID	Test Event Result	Action

This page intentionally left blank.

Bug Tracking Report
Bug's Report Title:
Steps Involved to Reproduce the Error:
Expected Result:
Actual Result:
Notes:

This page intentionally left blank.

Weekly Status Report
Status for: Week Ending:
This Week: 1. 2. 3. 4. 5.
Goals for Next Week: 1. 2. 3. 4. 5.
Issues: 1. 2. 3. 4. 5.

This page intentionally left blank.

Identification Log			
Ref #	Type of Issue	Priority	Description

This page intentionally left blank.

Resolution Log						
Ref #	Status	Last Action Date	Action	Parties Involved	Decision Made	Resolution

This page intentionally left blank.

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z

READ THE FOLLOWING BEFORE OPENING THE PACKAGE.

This software contains files to help you utilize the models described in the accompanying book. By opening the package, you are agreeing to be bound by the following agreement:

This software product is sold as is without warranty of any kind, either express or implied, including but not limited to the implied warranty of merchantability and fitness for a particular purpose. Neither Wiley nor its dealers or distributors assumes any liability for any alleged or actual damages arising from the use of or the inability to use this software. (Some states do not allow the exclusion of implied warranties, so the exclusion may not apply to you.)