Unit III

TESTING FOR SPECIALIZED ENVIRONMENT

- Testing Client /Serer Systems
- Testing in a Multiplatform Environment
- Testing Object Oriented Software – Object Oriented Testing
- Testing web based systems – Web Technology Evolution
- Traditional software and web based software
- Challenges in Testing for web based software
- Quality aspects
- Web Engineering
- Case study for Web application testing

Testing Client/ Server Systems

- The concerns about client/server systems reside in the area of control. The testers need to determine that adequate controls are in place to ensure accurate, complete, timely and secure processing of client/server software systems. The testers must address the following five concerns :
- 1. Organizational readiness : The culture is adequately prepared to process data using client/server technology. Readiness must be evaluated in the areas of management, client installation, and server support.
- 2. Client installation : The concern is that the appropriate hardware and software will be in place to enable processing that will meet client needs.
- 3. Security – There is need for protection of both the hardware, including residence software, and the data that is processed using that hardware and software. Security must address threats from employees, outsiders, and acts of nature.
- 4. Client data – Controls must be in place to ensure that everything is not lost, incorrectly processed, or processed differently on a client workstation than in other areas of the organization
- 5. Client / server standards – Standards must exist to ensure that all client workstations operate under the same set of rules.

Workbench

- The figure provides a workbench for testing client/server systems. This workbench can be used in steps as the client/server system is developed or concurrently after the client system has been developed.

The AdHoc Process

- The adhoc process level is unpredictable and often very chaotic. The organization operates without formalized procedures, cost estimates and project plans.
- Change control is relax and there is little senior management exposure or understanding of the problems and issues.
- Although organizations at this level may have formal procedures for planning and tracking their work, there is no management mechanism to ensure that they are used.
- One key reason why organizations behave in this fashion is that they have not experienced the benefits of a mature process and thus do not understand the consequences of their chaotic behavior.
- Organizations at the ad hoc process level can improve their performance by instituting basic project controls The most Important are project management , quality assurance and change control.
- Disciplined software development organization must have senior management oversight which includes
  - Review and approval of all major development plans
  - Quality review to be conducted
  - Installed quality performance
  - Schedule tracking, cost trends computing service and quality and productivity goals
- Quality assurance group is charged with assuring management that software work is done the way it is supposed to be done.
  - Like independent reporting line to senior management
  - sufficient resources to monitor performance of all key planning, implementation and verification activities.

Change control for software

- Change control for software is fundamental to business and financial control as well as to technical stability.
- If changes are not controlled, then orderly design, implementation, and testing are impossible and no quality plan can be effective.

The Repeatable Process

- It Provides control over the way the organization establishes its plans and commitments This control provides such an improvement over the adhoc process level
- Organizations at the repeatable process level face major risks when they are presented with new challenges.
- The following are some of the examples of the changes that represent the highest risk at this level
  - Unless the changes are introduced with great care, new tools and methods can negatively affect the testing process.
  - When the organization develop a new kind of product, it is entering new territory. The changes may eliminate lessons learned through experience
  - Major organizational changes also be highly disruptive. A new manager has no orderly basis for understanding the organization's operation, and new team members must learn the ropes through word of mouth.
- The key actions required is to establish a process group, establish a development process architecture and to introduce a software engineering methods and technologies
- These include design and code inspections, formal design methods, library control systems and comprehensive testing methods.
- Consistent Process
  - The organization has achieved the foundation for major and continuing progress.
  - There are some uncertainties related to lack of process definition and the consequent confusion about the specific items to be measured.
  - With a consistent process, measurements can be focused on specific tasks The process architecture is thus an essential prerequisite to effective management.

The following key steps are required to advance from consistent process level to the measured process level

- Establish a minimum basic set of process measurements to identify the quality and cost parameters of each process step. The objective is to quantify the relative costs and benefits of each major process activity
- Establish a process database and the resources to manage and maintain it. Cost and productivity data should be maintained centrally to guard against loss, to make it available for all projects, and to facilitate process quality and productivity analysis.
- Provide sufficient process resources to gather and maintain this process data and to advise project members on its use. Assign skilled professionals to monitor the quality of the data before entry in the database and to provide guidance on analysis methods and interpretation
- Assess the relative quality of each product and inform management where quality targets are not being met. An independent quality assurance group should assess the quality actions of each project and track its progress against its quality plan. When this progress is compared with the historical experience on similar projects, an informed assessment can generally be made.

The Measured Process

- In advancing from the ad hoc process through the repeatable and consistent processes to the measured process, software organizations should expect to make substantial quality improvements
  - There are an enormous number of potentially valuable measures of the software process, but such data is expensive to gather and to maintain.
  - For example, the simple measure of lines of source code per expended development month can vary by 100 times or more, depending on the interpretation of the parameters
  - Lines of code need to be defined to get consistent counts. . It is rare that two projects are comparable by any simple measures
  - Similarly, the cost per line of code of small modifications is often two to three times that for new programs

- The degree of requirements change can make an enormous difference.
- The two fundamental requirements for advancing from the measured process to the next level are:
- 1. Support automatic gathering of process data. All data is subject to error and omission, some data cannot be gathered by hand, and the accuracy of manually gathered data is often poor.
- 2. Use process data both to analyze and to modify the process to prevent problems and improve efficiency.

The Optimized Process

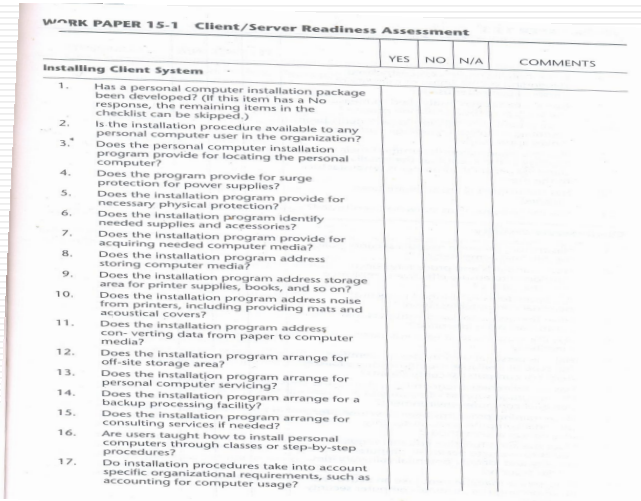
- With the step from the measured to the optimized process, however, there is a paradigm shift.
- Up to this point, software development managers have largely focused on their products and will typically gather and analyze only data that directly relates to product improvement
- In the optimized process, the data is available to tune the process itself.
- With a little experience, management will soon see that process optimization can produce major quality and productivity benefits

- With the optimized process, the organization has the means to identify the weakest elements of the process and to fix them.
- At this point in process improvement, data is available to justify the application of technology to various critical tasks, and numerical evidence is available on the effectiveness with which the process has been applied to any given product

Conducting the Client/Server Readiness Assessment

- To perform the client/server readiness assessment, we need to evaluate your organization in some readiness dimensions
- We may want to assemble a representative group of individuals from your organization to develop the assessment and use Work Paper as shown in the fig. to assist them in performing the assessment
- You should rate each readiness dimension in one of the following four categories:

- Motivation – level of commitment to use client /server technology
- Investment – Amount of money budgeted/allotted for expenditure of client/server program
- Client/server skills – the ability of the client/server installation team to incorporate client/server program
- User education – Awareness of the individuals involved in c/s technology
- Culture – The willingness of the organization to innovate
- Client/server support staff
- Client/server aids and tools
- Process maturity

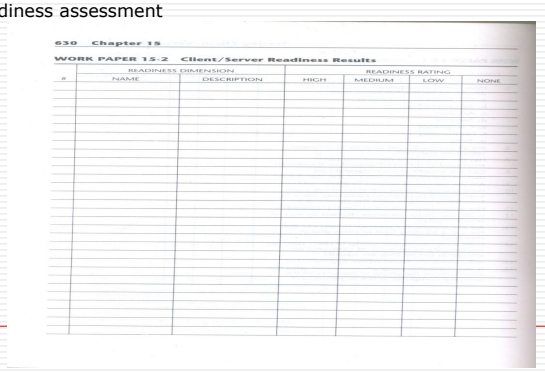


WORK PAPER 15-1 Client/Server Readiness Assessment

Installing Client System	YES	NO	N/A	COMMENTS
1. Has a personal computer installation package been developed? (If this item has a No response, the remaining items in the checklist can be skipped.)				
2. Is the installation procedure available to any personal computer user in the organization?				
3. Does the personal computer installation program provide for locating the personal computer?				
4. Does the program provide for surge protection for power supplies?				
5. Does the installation program provide for necessary physical protection?				
6. Does the installation program identify needed supplies and accessories?				
7. Does the installation program provide for acquiring needed computer media?				
8. Does the installation program address storing computer media?				
9. Does the installation program address storage area for printer supplies, books, and so on?				
10. Does the installation program address noise from printers, including providing mats and acoustical covers?				
11. Does the installation program address con- verting data from paper to computer media?				
12. Does the installation program arrange for off-site storage area?				
13. Does the installation program arrange for personal computer servicing?				
14. Does the installation program arrange for a backup processing facility?				
15. Does the installation program arrange for consulting services if needed?				
16. Are users taught how to install personal computers through classes or step-by-step procedures?				
17. Do installation procedures take into account specific organizational requirements, such as accounting for computer usage?				

(continues)

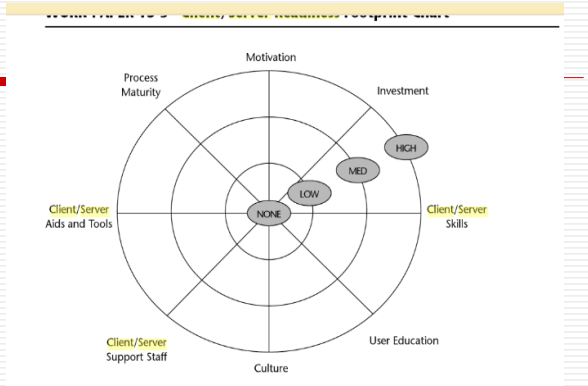
- 1. High. The readiness assessment team is satisfied that the readiness in this dimension will not inhibit the successful implementation of client/server technology.
- 2. Medium. The readiness assessment team believes that the readiness in this dimension will not be a significant factor in causing the client/server technology to fail. While additional readiness would be desirable, it is not considered an inhibitor to installing client/server technology.
- 3. Low. While there is some readiness for client/server technology, there are serious reservations that the readiness in this dimension will have a negative impact on the implementation of client/server technology.
- 4. None. No readiness at all in this area. Without at least low readiness in all eight dimensions, the probability of client/server technology being successful is extremely low. Work Paper 15-2 can be used to record the results of the client/server technology readiness assessment


630 Chapter 15

WORK PAPER 15-2 Client/Server Readiness Results

#	READINESS DIMENSION NAME	DESCRIPTION	READINESS RATING HIGH	MEDIUM	LOW	NONE

- Preparing a Client/Server Readiness Footprint Chart
  - A footprint chart is a means of graphically illustrating readiness. The end result will be a footprint that indicates the degree of readiness. The chart is completed by performing the following two steps
  - 1. Record the point on the dimension line that corresponds to the readiness rating provided on Work Paper 15-2. For example, if the motivation dimension was scored medium, place a dot on the medium circle where it intersects with the motivation dimension line.
  - 2. Connect all of the points and color the inside of the readiness lines connecting the eight readiness points.



Task 2: Assess Key Components

- Experience shows that if the key or driving components of technology are in place and working, they will provide most of the assurance necessary for effective processing. Four key components are identified for client/server technology
- 1. Client installations are done correctly.
- 2. Adequate security is provided for the client/server system
- 3. Client data is adequately protected
- 4. Client/server standards are in place and working
- These four key components need to be assessed prior to conducting the detailed testing. Experience has shown that if these key components are not in place and working, the correctness and accuracy of ongoing processing may be degraded even though the software works effectively.

- The checklists are used most effectively if they are answered after an assessment of the four key areas is completed. The questions are designed to be answered by the testers and not to be asked of the people developing the key component areas.

Task 3: Assess Client Needs

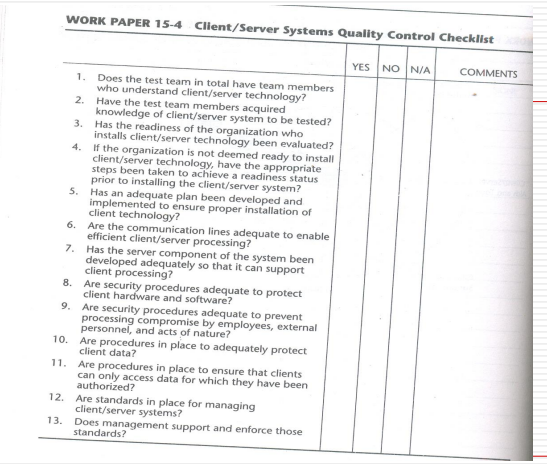
- Assessing client needs in a client/server system is a challenge because of the number of clients.
- In some organizations, clients will be homogenous, whereas in other organizations, they will be diverse and not in direct communication with one another
- The tester's challenge is that a client/server system that meets the needs of some clients might not meet the needs of all clients.
- Testers need some assurance that the client/server system incorporates the needs of the all clients.
- The tester has two major challenges in evaluating the needs of the clients
  - Can the system do what the client need
  - Can the client produce results consistent with other clients and other systems?

- The tester has two options in validating whether the client/server system will meet the processing needs of the clients
- Two distinct methods might be used.
  - The first is that the client/server system is developed for the clients
  - The second method is when the client/server system is built based on the requests of the clients. If the clients have specified their needs, it would be beneficial to conduct a review of the documented requirements for accuracy and completeness
- The following are some of the client/server characteristics that testers should evaluate to determine whether they meet the client needs:

- Data formats. The format in which the client receives data is a format that is usable by the client. This is particularly important when the client will be using other software systems to ensure that the data formats are usable by those other systems
  - Completeness of data. Clients may need more data to correctly perform the processing desired. For example, in our accounting cut-off discussion, it is important that the client would know in which accounting period the data belongs. In addition, there may be data needed by the user that is not provided by the client/server system.
  - Understandable documentation. The client needs to have documentation, both written and onscreen, that is readily understandable. For example, abbreviations should be clearly defined in the accompanying written documentation
  - Written for the competency of the client's users. Many software developers develop software for people at higher competency levels than the typical users of their systems. This can occur if the system is complex to use, or assumes knowledge and background information not typical of the clients that will use the system.

- Easy to use. Surveys have shown that only a small portion of most software systems is used because of the difficulty in learning how to use all components of the system. If a processing component is not easy to use, there is a high probability that the client will not use those parts of the system correctly
- Sufficient help routines. Periodically, clients will be involved in a processing situation for which they do not know how to respond. If the client/server system has "help" routines available, the probability that the client can work through those difficult situations is increased.

- **Check Procedures**
- Work Paper 2 is a quality control checklist for this client/server test process. It is designed so that Yes responses indicate good test practices, and No responses warrant additional investigation. A Comments column is provided to explain No responses and to record results of investigation. The N/A column is used when the checklist item is not applicable to the test situation



WORK PAPER 15-4 Client/Server Systems Quality Control Checklist

	YES	NO	N/A	COMMENTS
1. Does the test team in total have team members who understand client/server technology?				
2. Have the test team members acquired knowledge of client/server system to be tested?				
3. Has the readiness of the organization who installs client/server technology been evaluated?				
4. If the organization is not deemed ready to install client/server technology, have the appropriate steps been taken to achieve a readiness status prior to installing the client/server system?				
5. Has an adequate plan been developed and implemented to ensure proper installation of client technology?				
6. Are the communication lines adequate to enable efficient client/server processing?				
7. Has the server component of the system been developed adequately so that it can support client processing?				
8. Are security procedures adequate to protect client hardware and software?				
9. Are security procedures adequate to prevent processing compromise by employees, external personnel, and acts of nature?				
10. Are procedures in place to adequately protect client data?				
11. Are procedures in place to ensure that clients can only access data for which they have been authorized?				
12. Are standards in place for managing client/server systems?				
13. Does management support and enforce those standards?				

- **Output**
- The only output from this system is the test report indicating what works and what does not work. The report should also contain recommendations by the test team for improvements, where appropriate

- **Testing in a Multiplatform Environment**
- Software designed to run on more than one platform must undergo two tests
- The first test is to validate that the software performs its intended functions.
- The second test is that the software will perform in the same manner regardless of the platform on which it is executed
- This chapter provides a six-task process for testing in a multiplatform environment.
- The test process presumes that the platforms for which the software must execute are known

- The process also presumes that the software has already been tested and that testers have validated that it performs its intended functions correctly.

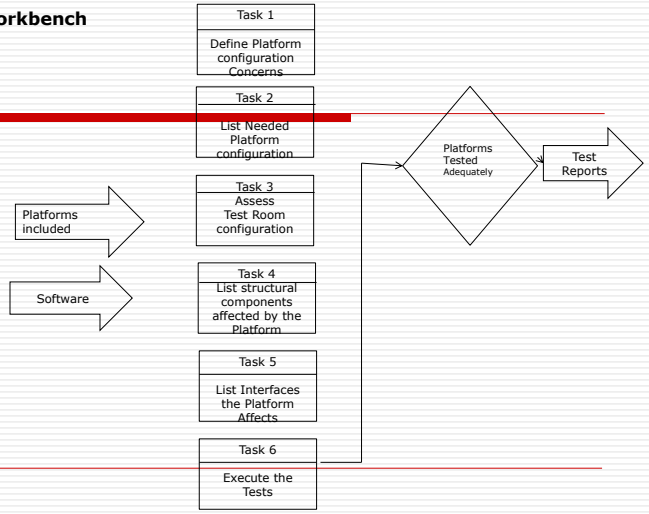
- **Overview**
- Each platform on which software is designed to execute operationally may have slightly different characteristics.
- These distinct characteristics include various operating systems, hardware configurations, operating instructions, and supporting software, such as database management systems.
- These different characteristics may or may not cause the software to perform its intended functions differently.

- The objective of testing is to determine whether the software will produce the correct results on various platforms

- **Objective**
- The objective of this six-task process is to validate that a single software package executed on different platforms will produce the same results
- The test process is basically the same as was used in parallel testing.
- Software must operate on multiple platforms with the individual results being compared to ensure consistency in output.
- The testing normally requires a test lab that includes the predetermined platforms

- **Concerns**
- The following are the three major concerns for testing in a multiplatform environment
- 1. The platforms in the test lab will not be representative of the platforms in the real world. This can happen because the platform in the test lab may not be upgraded to current specifications, or it may be configured in a manner that is not representative of the typical configuration for that platform.
- 2. The software will be expected to work on platforms not included in the test labs. By implication, users may expect the software to work on a platform that has not been included in testing.
- 3. The supporting software on various platforms is not comprehensive. User platforms may contain software that is not the same as that used on the platform in the test lab (for example, a different database management system

Workbench



- Figure illustrates the workbench for testing in a multiplatform environment. This figure shows that six tasks are needed to effectively test in a multiplatform environment
- Most tasks assume that the platforms will be identified in detail, and that the software to run on the different platforms has been previously validated as being correct
- Five of the six tasks are designed to determine what tests are needed to validate the correct functioning of the identified platforms, and the sixth task executes those tests.

- **Input**
- The two inputs for testing in a multiplatform environment are as follows
- 1.List of platforms on which software must execute. The main requirement for multiplatform testing is a list of the platforms. These platforms must be described in detail as input to testing or described in detail prior to commencing testing
- 2. Software to be tested. The software package(s) to be tested is input to the test process. This software must be validated that it performs its functions correctly prior to multiplatform testing. If this has not been done, then the software should be subject to the seven-step testing process, described in Part Three of this book, prior to commencing multiplatform testing.

- **Do Procedures**

The following six tasks should be performed to validate that software performs consistently in a multiplatform environment:

1. Define platform configuration concerns.
2. List needed platform configurations.
3. Assess test room configurations.
4. List structural components affected by the platform(s).
5. List interfaces platform affects.
6. Execute the tests.

- **Task 1: Define Platform Configuration Concerns**
- The first task in testing a multiplatform environment is to develop a list of potential concerns about that environment
- The testing that follows will then determine the validity of those concerns.
- The recommended process for identifying concerns is error guessing
- so the software test experts can predict the types of defects that will occur in software. This means that the types of problems that you encounter in one will occur in most other similar tests

- For example, the problem of data exceeding its allocated field size will appear sooner or later in almost all software applications.
- If you anticipate it and decide what you will do when it happens and how the software will react to the situation, successful use of the software will not be threatened.
- Error guessing requires the following two prerequisites:
  1. The error-guessing group understands how the platform works.
- 2. The error-guessing group knows how the software functions.

- The following is a short list of questions to brainstorm during error guessing
- Does the software have any unusual transactions?
- What are the most common errors that you are now making
- What would happen to processing if you forgot to perform one of the step
- What would happen if you did not enter all of the data in an input transaction?
- Will you be able to determine who performed what computer operation in case questions arise regarding the correctness of operations?
- ■■ If a diagnostic message is produced by the computer, how will you know it has been properly corrected

- How will you know the person operating the computer knows how to operate it correctly?
- These questions are designed to spark ideas about what might go wrong within a platform.

- **Task 2: List Needed Platform Configurations**
- The test must identify the platforms that must be tested
- this list of platforms and detailed description of the platforms would be input to the test process
- The needed platforms are either those that will be advertised as acceptable for using the software, or platforms within an organization on which the software will be executed.
- The platforms need to be described in detail.
- Testers must then determine whether those platforms are available for testing
- If the exact platform is not available, the testers need to determine whether an existing platform is acceptable

- For example, if an available platform did not contain some feature or configuration, would the existing platform provide a reasonable test?
- If so, that platform can be used for testing. If the needed platform is not available, the testers must make a determination of whether to obtain such a platform or accept the risk that the software will be released without testing that specific platform
- The determination of whether an available test platform meets the needed test platform should be recorded

- **Task 3: Assess Test Room Configurations**
- The testers need to determine whether the platforms available in the test room are acceptable for testing.
- This involves the following two steps
- : 1. For each needed platform listed on Work Paper document the platform to be used for testing, if any is available, on the Work Paper
- 2. Make a determination as to whether the available platform is acceptable for testing. Indicate your decision on Work Paper 19-1.
- If the platform is not acceptable, note any actions to be taken.

- **Task 4: List Structural Components Affected by the Platform(s)**
- Structural testing deals with the architecture of the system
- Architecture describes how the system is put together. It is used in the same context that an architect designs a building. Some of the architectural problems that could affect computer processing include: Testing in a Multiplatform Environment
  - ■■ Internal limits on the number of events that can occur in a transaction (for example, the number of products that can be included on an invoice
  - Maximum size of fields (for example, the quantity is only two positions in length, making it impossible to enter an order for more than 99 items

- Disk storage limitations (for example, you are permitted to have only X customers).
- Performance limitations (for example, the time to process transactions jumps significantly when you enter more than X transactions
- The questions at hand are:
- Do you feel competent to do it?
- Is it worth doing?
- The answers to these questions depend on the critical nature of the software and what would happen if your business was unable to continue computer processing because you reached the program limitation

- Structural testing also relates to file-handling problems. Such file problems include incorrect processing when the last record on file is updated or adding a record that will become the first record on a file. In the personal computer software market, literally thousands of people are writing software
- As an aid in developing structural test conditions, the more common structural problem areas are listed in the following text
- Adding a record before the first record on a file
- Adding a record after the last record on a file.
- Deleting the first record on a file
- Deleting the last record on a file
- Changing information on the first record on a file
- Changing information on the last record on a file

- Causing the program to terminate by predetermined conditions
- Accumulating a field larger than the mathematical accumulators can hold
- Verifying that page counters work
- Verifying that page spacing works
- Entering invalid transaction types
- Entering invalid values in fields
- Processing unusual conditions
- Testing major error conditions
- Testing for out-of-control conditions (for example, whether the value of the records in the batch do not equal the entered batch total)

- Simulating a hardware failure that forces recovery procedures to be used
- Demonstrating recovery procedures
- Entering more records than disk storage can hold
- Entering more values than internal tables can hold
- ■■ Entering incorrect codes and transaction types
- Entering unreasonable values for transaction processing
- Violating software rules not violated by preceding structural test conditions

- **Check Procedures**
- Prior to completing multiplatform testing, a determination should be made that testing was performed correctly
- Work Paper provides a series of questions to challenge the correctness of multiplatform testing
- A Yes response to those items indicates that multiplatform testing was performed correctly;
- a No response indicates that it may or may not have been done correctly. Each No response should be clarified in the Comments column
- The N/A column is for items that are not applicable to this specific platform test.

- **Output**
- The output from this test process is a report indicating the following
- Structural components that work or don't work by platform
- Interfaces that work or don't work by platform
- Multiplatform operational concerns that have been eliminated or substantiated
- Platforms on which the software should operate but that have not been tested
- The report will be used to clarify the software's operating instructions and/or make changes to the software

Testing Web-Based Systems

- Web-based architecture is an extension of client/server architecture
- In a client/server architecture, as discussed software resides on the client workstations.
- For web-based systems, the browsers reside on client workstations. These client workstations are networked to a web server, either through a remote connection or through a network such as a local area network (LAN) or wide area network (WAN).

☐ As the web server receives and processes requests from the client workstation, requests may be sent to the application server to perform actions such as data queries, electronic commerce transactions, and so forth.

☐For example, when an online banking transaction is processed over the Internet, the transaction is eventually updated to the customer's account and shown on a statement in a back-end process.

☐ **Concerns**

☐Testers should have the following concerns when conducting web-based testing:

☐**Browser compatibility**. Testers should validate consistent application performance on a variety of browser types and configurations.

☐**Functional correctness**. Testers should validate that the application functions correctly. This includes validating links, calculations, displays of information, and navigation.

☐**Integration.** Testers should validate the integration between browsers and servers, applications and data, and hardware and software.

☐ **Usability**. Testers should validate the overall usability of a web page or a web application, including appearance, clarity, and navigation.

☐**Security**. Testers should validate the adequacy and correctness of security controls, including access control and authorizations.

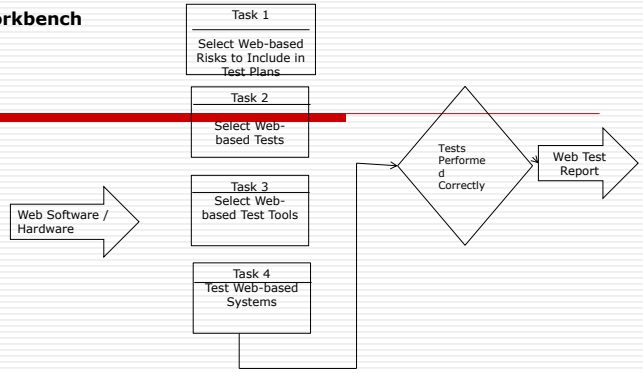
☐**Performance**. Testers should validate the performance of the web application under load.

☐**Verification of code**. Testers should validate that the code used in building the web application (HTML, Java, and so on) has been used in a correct manner. For example, no nonstandard coding practices should be used that would cause an application to function incorrectly in some environments.

Workbench

- The input to the workbench is the hardware and software that will be incorporated in the web-based system to be tested.
- The first three tasks of the workbench are primarily involved in web-based test planning.
- The fourth task is traditional software testing.
- The output from the workbench is to report what works and what does not work, as well as any concerns over the use of web technology

Workbench



❑ The following list shows how web-based systems differ from other technologies. The description of the web-based systems under testing should address these differences:

❑ **Uncontrolled user interfaces** (browsers). web page must be functional on those browsers that you expect to be used in accessing your web applications. Furthermore, as new releases of browsers emerge, your web applications will need to keep up with compatibility issues.

❑ **Complex distributed systems.** web-based applications are also remotely accessed, which adds even more concerns to the testing effort. While some applications may be less complex than others, it is safe to say that the trend in web applications is to become more complex rather than less

☐**Security issues**. Protection is needed from unauthorized access that can corrupt applications and/or data. Another security risk is that of access to confidential information

☐**Multiple layers in architecture**. These layers of architecture include application servers, web servers, back-end processing, data warehouses, and secure servers for electronic commerce.

❑ **New terminology and skill sets**. Just as in making the transition to client/ server, new skills are needed to develop, test, and use web-based technology effectively.

❑ **Object-oriented**. Object-oriented languages such as Java are the mainstay of web development.

Task 1: Select Web-Based Risks to Include in the Test Plan

❑ Risks are important to understand because they reveal what to test.

❑ Each risk points to an entire area of potential tests.

❑ In addition, the degree of testing should be based on risk.

❑ The risks are briefly listed here, followed by a more detailed description of the concerns associated with each risk:

❑ **Security**. As we have already seen, one of the major risks of Internet applications is security. It is very important to validate that the application and data are protected from outside intrusion or unauthorized access.

- Performance. An Internet application with poor performance will be judged hard to use. Web sites that are slow in response will not retain the visitors they attract and will be frustrating to the people who try to use them.
- Correctness. Obviously, correctness is a very important area of risk. It is essential that the functionality and information obtained from web-based applications are correct.

- **Compatibility** (configuration). A web-based application must be able to work correctly on a wide variety of system configurations including browsers, operating systems, and hardware systems. All of these are out of the control of the developer of the application.
- **Reliability**. An Internet application must have a high level of availability and the information provided from the application must be consistent and reliable to the user.
- **Data integrity**. The data entered into an Internet application must be validated to ensure its correctness. In addition, measures must be taken to ensure the data stays correct after it is entered into the application.

- **Usability**. The application must be easy to use. This includes things like navigation, clarity, and understandability of the information provided by the application.
- **Recoverability**. In the event of an outage, the system must be recoverable. This includes recovering lost transactions, recovering from loss of communications, and ensuring that proper backups are made as a part of regular systems maintenance.

- Security Concerns
- **External intrusion**. Perhaps the most obvious security concern is that of protecting the system from external intrusion. This can include intrusion from people who are trying to gain access to sensitive information, people who are trying to intentionally sabotage information, and people who are trying to intentionally sabotage applications.
- **Protection of secured transactions**. Another major area of concern is that of protecting transactions over the Internet. This is especially true in dealing with electronic commerce transactions. Many consumers are reluctant to give credit card information over the Internet for fear that information will be intercepted and used for fraudulent purposes

- **Viruses**. The Internet has become a vehicle for propagating tens of thousands of new viruses. These viruses are contained in downloaded files that can be distributed from web sites and e-mail.
- **Access control**. Access control means that only authorized users have security access to a particular application or portion of an application. This access is typically granted with a user ID and password.
- **Authorization levels**. Authorization levels refer to the ability of the application to restrict certain transactions only to those users who have a certain level of authorization.

- **Performance Concerns**
- Performance testing is a very precise kind of testing and requires the use of automated tools for testing to be accomplished with any level of accuracy and efficiency
- Manual approaches to performance testing fall short of the accuracy needed to correctly gauge an application's performance
- Load testing seeks to determine how the application performs under expected and greater-than-expected levels of activity
- Application load can be assessed in a variety of ways:

- **Concurrency**. Concurrency testing seeks to validate the performance of an application with a given number of concurrent interactive users
- **Stress**. Stress testing seeks to validate the performance of an application when certain aspects of the application are stretched to their maximum limits. This can include maximum number of users, and can also include maximizing table values and data values.
- **Throughput**. Throughput testing seeks to validate the number of transactions to be processed by an application during a given period of time. For example, one type of throughput test might be to attempt to process 100,000 transactions in one hour.

- **Correctness Concerns**
- Of course, one of the most important areas of concern is that the application functions correctly. This can include not only the functionality of buttons and "behind the scenes" instructions but also calculations and navigation of the application.

- **Functionality**. Functional correctness means that the application performs its intended tasks as defined by a stated set of specifications. The specifications of an application are the benchmark of what the application should do. Functional correctness is determined by performing a functional test. A functional test is performed in a cause-effect manner. In other words, if a particular action is taken, a particular result should be seen.
- **Calculations**. Many web-based applications include calculations. These calculations must be tested to ensure correctness and to find defects

- Navigation. Navigation correctness can include testing links, buttons, and general navigation through a web site or web-based application.

- **Compatibility Concerns**
- Compatibility is the capability of the application to perform correctly in a variety of expected environments
- *Browser Configuration*
- The most reasonable testing strategy is to define optimal configurations on the most standard kinds of browsers and test based on those configurations.

- **Monitors, video cards, and video RAM**. If you have a web site that requires a high standard of video capability, some users will not be able to view your site, or will not have a positive experience at your site
- **Audio, video, and multimedia support**. Once again, you need to verify that a web application is designed to provide a level of multimedia support that a typical end-user will need to be able to access your site.
- **Memory (RAM) and hard drive space**. RAM is very important for increasing the performance of a browser on a particular platform. Browsers also make heavy use of caching, which is how a browser stores graphics and other information on a user's hard drive. This helps speed the display of web pages the next time the user visits a web site.

- **Bandwidth access**. Many corporate users have high-speed Internet access based on T-1 or T-3 networks, or ISDN telephone lines.
- Browser differences can make a web application appear differently to different people. These differences may appear in any of the following areas
- **Print handling**. To make printing faster and easier, some pages add a link or button to print a browser-friendly version of the page being viewed.
- **Reload**. Some browser configurations will not automatically display updated pages if a version of the page still exists in the cache. Some pages indicate if the user should reload the page.

- **Navigation**. Browsers vary in the ease of navigation, especially when it comes to visiting pages previously visited during a session. A web application developer may need to add navigational aids to the web pages to facilitate ease of navigation.
- **Graphics filters**. Browsers may handle images differently, depending on the graphic filters supported by the browser. In fact, some browsers may not show an image at all. By standardizing on JPG and GIF images you should be able to eliminate this concern.
- **Caching**. How the cache is configured (size, etc.) will have an impact on the performance of a browser to view information.

- Dynamic page generation. This includes how a user receives information from pages that change based on input. Examples of dynamic page generation include:
  - Shopping cart applications
  - Data search applications
  - Calculation forms
  - File downloads. Movement of data from remote data storage for user processing.
  - E-mail functions. Because e-mail activities can consume excessive processing time, guidelines should be developed.
- **Reliability Concerns** Because of the continuous uptime requirements for most Internet applications, reliability is a key concern. However, reliability can be considered in more than system availability; it can also be expressed in terms of the reliability of the information obtained from the application

- **Data Integrity Concerns** Not only must the data be validated when it is entered into the web application, but it must also be safeguarded to ensure the data stays correct:
- **Ensuring only correct data is accepted**. This can be achieved by validating the data at the page level when it is entered by a user.
- **Ensuring data stays in a correct state**. This can be achieved by procedures to back up data and ensure that controlled methods are used to update data.

- **Usability Concerns** If users or customers find an Internet application hard to use, they will likely go to a competitor's site. Usability can be validated and usually involves the following:
  - Ensuring the application is easy to use and understand
  - Ensuring that users know how to interpret and use the information delivered from the application
  - Ensuring that navigation is clear and correct
- **Recoverability Concerns** Internet applications are more prone to outages than systems that are more centralized or located on reliable, controlled networks. The remote accessibility of Internet applications makes the following recoverability concerns important:

- Lost connections
- Timeouts
- Dropped lines
- Client system crashes
- Server system crashes or other application problems

WORK PAPER 22-1 Web-Based Risks to Include in the Test Plan

Field Requirements

FIELD	INSTRUCTIONS FOR ENTERING DATA
Web-based Risks	This field lists the eight web-based risks described in this chapter. The description implies that "lack of" is associated with the risk.
Include in Test	The web-based testing should determine whether any or all of the eight identified web-based risks need to be addressed in the test plan. A check in the Yes column indicates that it should be included in the plan, and a check in the No column indicates it is not needed in the plan.
How risk will be included in the web-based test plan	This column is designed to be used in two ways. If the risk is not to be included in test plan, a justification as to why not could be included in this column. The second use is the test team's preliminary thoughts on how this risk will be included in the test plan. The description might involve the types of tests, the types of tools, and/or the approach to be used in testing.

WEB-BASED RISKS (LACK OF)	INCLUDE IN TEST		HOW RISK WILL BE INCLUDED IN WEB-BASED TEST PLAN
	YES	NO	
Security			
Performance			
Correctness			
Compatibility (Configuration)			
Reliability			
Data Integrity			
Usability			
Recoverability			

- Task 2: Select Web-Based Tests
- Unit or Component
  - This includes testing at the object, component, page, or applet level. Unit testing is the lowest level of testing in terms of detail. During unit testing, the structure of languages, such as HTML and Java, can be verified. Edits and calculations can also be tested at the unit level.
- Integration
  - Integration is the passing of data and/or control between units or components, which includes testing navigation (i.e., the paths the test data will follow). In web-based applications, this includes testing links, data exchanges, and flow of control in an application.

- System
  - System testing examines the web application as a whole and with other systems. The classic definition of system testing is that it validates that a computing system functions according to written requirements and specifications.
  - This is also true in web based applications. The differences apply in how the system is defined. System testing typically includes hardware, software, data, procedures, and people.
  - In corporate web-based applications, a system might interface with Internet web pages, data warehouses, back-end processing systems, and reporting systems.

- User Acceptance
  - This includes testing that the web application supports business needs and processes. The main idea in user acceptance testing (or business process validation) is to ensure that the end product supports the users' needs.
  - For business applications, this means testing that the system allows the user to conduct business correctly and efficiently.
  - For personal applications, this means that users are able to get the information or service they need from a web site efficiently.

- In a corporate web page, the end-user testers may be from end-user groups, management, or an independent test team that takes the role of end users.
- In public web applications, the end-user testers may be beta testers, who receive a prototype or early release of the new web application, or independent testers who take the role of public web users

- Performance
  - This includes testing that the system will perform as specified at predetermined levels, including wait times, static processes, dynamic processes, and transaction processes. Performance is also tested at the client/browser and server levels.
- Load/Stress
  - This type of testing checks to see that the server performs as specified at peak concurrent loads or transaction throughput. It includes stressing servers, networks, and databases.
- Regression
  - Regression testing checks that unchanged parts of the application work correctly after a change has been made.
  - The main idea is to test a set of specified critical test cases each time you perform the test. Regression testing is an ideal candidate for test automation because of its repetitive nature.

- Usability
  - This type of testing assesses the ease of use of an application. Usability testing may be accomplished in a variety of ways, including direct observation of people using web applications, usability surveys, and beta tests. The main objective of usability testing is to ensure that an application is easy to understand and navigate.
- Compatibility
  - Compatibility testing ensures that the application functions correctly on multiple browsers and system configurations. Compatibility testing may be performed in a test lab that contains a variety of platforms, or may be performed by beta testers.

- The downside with beta testing is the increased risk of bad publicity, the lack of control, and the lack of good data coming back from the beta testers.
- Work Paper 2 is designed to assist testers in selecting testing types. The type of testing to be performed should be focused on the web-based risks addressed by the test plan.
- The test team should determine how the various types of web-based testing selected should be used to assess the various risks.
- This work paper, like Work Paper 22-1, should be developed through brainstorming and consensus by the web-based test team.

WORK PAPER 22-2 Types of Web-Based Testing to Perform

Field Requirements

FIELD	INSTRUCTIONS FOR ENTERING DATA
Types of Web-based Testing	This column contains the more common types of web-based testing. The names may need to be modified for your culture. Additional types of testing performed by your test group may need to be added to this column.
Perform	This field is used for the web-based test team to indicate which types of testing will be used during web-based testing. A check mark in the Yes column indicates the type of testing that will be performed, and check mark in the No column indicates that type of testing will not be performed.
Risk Focus	The web-based test team should indicate the risk that this test type will be used to address. The type of risk to be incorporated into the test plan has been identified on Work Paper 22-1. In addition, the column can be used to indicate the justification for not using various types of web-based testing, if appropriate.
How to Be Used	The web-based test team should write a brief narrative description of how they plan to use this test type to address the risks that will be incorporated into the test plan.

TYPES OF WEB-BASED TESTING	PERFORM		RISK FOCUS	HOW TO BE USED
	YES	NO		
Unit/Component				
Integration				
System				
User Acceptance				
Performance				
Load/Stress				
Regression				
Usability				
Compatibility				

- Task 3: **Select Web-Based Test Tools**
- Effective web-based testing necessitates the use of web-based testing tools. A brief description of categories of the more common web-based test tools follows:
- ■■ **HTML tools**. Although many web development packages include an HTML checker, there are ways to perform a verification of HTML if you do not use/ have such a feature.
- ■■ **Site validation tools**. Site validation tools check your web applications to identify inconsistencies and errors, such as moved or orphaned pages and broken links.
- ■■ **Load/stress testing tools**. Load/stress tools evaluate web-based systems when subjected to large volumes of data or transactions.
- ■■ **Test case generators**. Test case generators create transactions for use in testing. This tool can tell you what to test, as well as create test cases that can be used in other test tools.

- Work Paper 3 is designed to document the web-based test tools selected by the test team, as well as how those tools will be used. The work paper should contain all of the specific test tools available to the web-based testing team.

WORK PAPER 22-3 Select Web-Based Test Tools

Field Requirements

FIELD	INSTRUCTIONS FOR ENTERING DATA
Web-based Test Tool	All of the test tools available to your web-based test team should be listed in this column. The column contains generic types of test tools, but they should be replaced by specific test tools.
Perform	The web-based test team should identify which web-based test tool will be used during testing. A check in the Yes column indicates that the tool is to be used, and check in the No column indicates that the tool is not to be used.
Test Type Focus	The test team should indicate in this column which type of testing will be performed using this test tool. The test types are those indicated by the check mark in the Yes column on Work Paper 22-3. All of the test types with a Yes check mark on Work Paper 22-2 should be addressed in this column. Note that a single test tool may be used for multiple test types.
How to Be Used	The web-based test team should indicate in this test column how they plan to use a specific test tool during web-based testing. The testers should be as specific as possible in completing this column.

	PERFORM			
WEB-BASED TEST TOOLS	YES	NO	TEST TYPE FOCUS	HOW TO BE USED
HTML text tool				
Site validation test tool				
Java test tool				
Load/stress test tool				
Test case generator				
Other (list tools)				

- Task 4: **Test Web-Based Systems**
- The tests to be performed for web-based testing will be the types of testing described in the seven-step testing process
- seven-step process may have to be modified based on the risks associated with web-based testing.

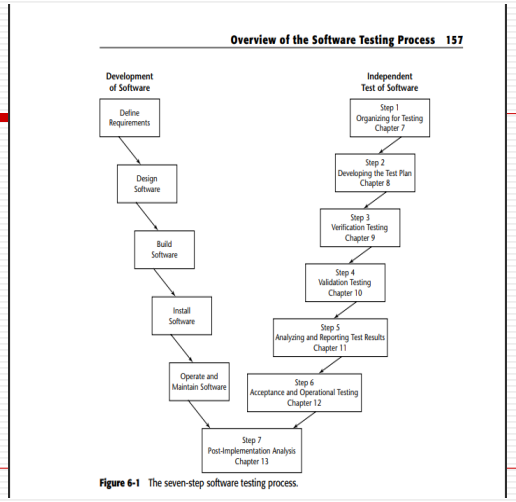


Figure 6-1 The seven-step software testing process.

- **Check Procedures**
  - The web-based test team should use Work Paper 4 to verify that the web-based test planning has been conducted effectively. The Comments column is provided to clarify No responses. The N/A column is provided for items that are not applicable to this specific web-based test plan.

WORK PAPER 22-4 Web-Based Testing Quality Control Checklist

		YES	NO	N/A	COMMENTS
1.	Has a web-based test team been organized?				
2.	Does the web-based test team understand the differences between client/server and web-based technology?				
3.	Does the web-based test team understand web terminology?				
4.	Does the web-based test team understand the risk associated with web technology?				
5.	Has the web-based test team reached consensus on which risks are applicable to this specific web-based system?				
6.	Has a determination been made as to how the identified risks will be incorporated in the test plan?				
7.	Is there a consensus that the web-based risks not included in the test plan are of minimal concern to this web-based system?				
8.	Has the web-based test team identified the types of testing required for this system?				
9.	If so, how have those testing types been correlated to the web-based risks?				
10.	Has the web-based test team reached consensus on how the web-based types of testing will be used for test purposes?				
11.	Is there a portfolio of web-based test tools available in the organization?				
12.	Are the available test tools adequate for the web-based system being tested?				
13.	Has each type of testing that will be included in the test plan been supported by a specific web-based test tool?				
14.	Has the test team reached consensus on how the test tools will be used during testing?				
15.	Have all of the web-based testing decisions made by the test team been incorporated into the test plan?				

- **Output**
- The only output from this test process is a report on the web-based system. At a minimum, this report should contain the following:
- A brief description of the web-based system
- The risks addressed and not addressed by the web-based test team
- The types of testing performed, and types of testing not performed
- The tools used
- The web-based functionality and structure tested that performed correctly
- The web-based structure and functionality tested that did not perform correctly