CHAPTER 14

Acceptance Testing

> Acceptance of others, their looks, their behaviors, their beliefs, bring you an inner peace and tranquility—instead of anger and resentment.
> — *Anonymous*

14.1 TYPES OF ACCEPTANCE TESTING

A product is ready to be delivered to the customer after the system test group is satisfied with the product by performing system-level tests. Customers execute acceptance tests based on their expectations from the product. The services offered by a software product may be used by millions of users. For example, the service provider of a cellular phone network is a customer of the software systems running the phone network, whereas the general public forms the user base by subscribing to the phone services. It is not uncommon for someone to have a dual role as a customer and a user. The service provider needs to ensure that the product meets certain criteria before the provider makes the services available to the general public. Acceptance testing is a formal testing conducted to determine whether a system satisfies its acceptance criteria—the criteria the system must satisfy to be accepted by the customer. It helps the customer to determine whether or not to accept the system [1]. The customer generally reserves the right to refuse to take delivery of the product if the acceptance test cases do not pass. There are two categories of acceptance testing:

- User acceptance testing.
- Business acceptance testing.

The UAT is conducted by the customer to ensure that system satisfies the contractual acceptance criteria before being signed off as meeting user needs. Actual planning and execution of the acceptance tests do not have to be undertaken directly by the customer. Often third-party consulting firms offer their services to do this task. However, the customer must specify the acceptance criteria for the third party to seek in the product. The BAT is undertaken within the development organization

Software Testing and Quality Assurance: Theory and Practice, Edited by Kshirasagar Naik and Priyadarshi Tripathy

of the supplier to ensure that the system will eventually pass the UAT. It is a rehearsal of UAT at the premises of the supplier. The development organization of the supplier derives and executes test cases from the client's contract, which include the acceptance criteria.

The acceptance criteria must be defined and agreed upon between the supplier and the customer to avoid any kind of protracted arguments. Either party or a third-party consulting firm may design the acceptance test plan. The acceptance criteria document is a part of the contract in the case of an outsourced development under the OEM agreement. If some hardware is an integral part of the system, then the hardware acceptance criteria are included in the contractual agreement. In general, the marketing organization of the buyer defines the acceptance criteria. However, it is important that the software quality assurance team of the buyer's organization initiate a dialogue with the seller and provide a set of "straw man" acceptance criteria for the marketing department to review and react to. The users, the system engineers, customer support engineers, and the software quality assurance group of the buyer's organization do the actual planning and execution of the acceptance tests after the criteria are agreed upon. The personnel developing an acceptance test plan must have a thorough understanding of the acceptance criteria that have been agreed upon. It is unlikely that the system passes all the acceptance criteria in one go for large, complex systems. It is useful to focus on the following three major objectives of acceptance testing for pragmatic reasons:

- Confirm that the system meets the agreed-upon criteria. The broad categories of criteria are explained in Section 14.2.
- Identify and resolve discrepancies, if there are any. The sources of discrepancies and mechanisms for resolving them have been explained in Section 14.5.
- Determine the readiness of the system for cut-over to live operations. The final acceptance of a system for deployment is conditioned upon the outcome of the acceptance testing. The acceptance test team produces an acceptance test report which outlines the acceptance conditions. The details of an acceptance test report have been explained in Section 14.6.

Acceptance testing is only one aspect of the contractual fulfillment of the agreement between a supplier and a buyer. A contractual agreement may require the seller to provide other materials, such as the design solution document that addresses the requirement document of the buyer. The acceptance test team may evaluate the acceptability of the system design in terms of graphical user interface, error handling, and access control.

14.2 ACCEPTANCE CRITERIA

At the core of any contractual agreement is a set of acceptance criteria. A key question is what criteria must the system meet in order to be acceptable? The acceptance criteria must be measurable and, preferably, quantifiable. The basic

principle of designing the acceptance criteria is to ensure that the quality of the system is acceptable. One must understand the meaning of the quality of a system, which is a complex concept. It means different things to different people, and it is highly context dependent [2].

Even though different persons may have a different view about quality, it is the customer's opinion that prevails. The concept of quality is, in fact, complex and multifaceted [3]. Five views of quality, namely, *transcendental view*, *user view*, *manufacturing view*, *product view*, and *value-based view*, have been explained in Chapter 17. The five views were originally presented by Garvin [3] in the context of production and manufacturing in general and subsequently explained by Kitchenham and Pfleeger [2] in a software development context. The five views are presented below in a concise form:

1. The transcendental view sees quality as something that can be recognized but is difficult to describe or define.
2. The user view sees quality as satisfying the purpose.
3. The manufacturing view sees quality as conforming to the specification.
4. The product view ties quality with the inherent characteristics of the product.
5. The value-based view puts a cost figure on quality—the amount a customer is willing to pay for it.

Acceptance criteria are defined on the basis of these multiple facets of quality attributes. These attributes determine the presence or absence of quality in a system. Buyers, or Customers, should think through the relevance and relative importance of these attributes in their unique situation at the time of formulating the acceptance criteria. The attributes of quality are discussed below and examples of acceptance criteria for each quality attribute are given.

Functional Correctness and Completeness One can ask the question: Does the system do what we want it to do? All the features which are described in the requirements specification must be present in the delivered system. It is important to show that the system works correctly under at least two to three conditions for each feature as a part of acceptance.

One can show the functional correctness of a system by using the requirements database, as discussed in Chapter 11. The database is used in generating a requirement traceability matrix during system-level testing. Basically a traceability matrix tells us the test cases that are used to verify a requirement and all the requirements that are partially verified by a test case. Such a traceability matrix is a powerful tool in showing the customer about the functional correctness of the system. It is important to obtain an early feedback from the customer on the requirements traceability matrix. The idea behind the feedback is to reach an agreement on the validation method to be employed for each requirement. The decision is especially significant because some validation methods are easier to implement

and less time intensive than other methods. For example, the demonstration method is less time intensive than the testing method.

In reality, rigorous functional correctness testing is conducted during the system testing phase, rather than during acceptance testing. However, the buyer may ask for the requirement traceability matrix before the start of acceptance testing to ensure that the system does function according to the requirement specification.

Accuracy The question is: Does the system provide correct results? Accuracy measures the extent to which a computed value stays close to the expected value. Accuracy is generally defined in terms of the magnitude of the error. A small gap—also called an error in numerical analysis, for example—between the actual value computed by a system and the expected value is generally tolerated in a continuous space. The accuracy problem is different in discrete space, leading to false-positive and false-negative results. False positives and false negatives are serious drawbacks in any diagnostic and monitoring software tools.

Data Integrity Data integrity refers to the preservation of the data while it is transmitted or stored such that the value of data remains unchanged when the corresponding receive or retrieve operations are executed at a later time. Thus, data must not be compromised by performing update, restore, retrieve, transmit, and receive operations. The requirement of data integrity is included in the acceptance test criteria to uncover design flaws that may result in data corruption. In communication systems, an intruder can change the data without the sender and receiver detecting the change. If integrity check mechanisms are in place, the data may be changed, but the mechanism will detect the tampering. Data integrity mechanisms detect changes in a data set. The concepts of message digest and digital signature are used in preserving data integrity [4].

Remark. A *message digest* algorithm takes in an input message of arbitrary length and produces a fixed-length code. The fixed-length code is called a *digest* of the original message. The commonly used message digest algorithms are the Message Digest 5 (MD5) and the Secure Hash Algorithm 1 and 2 (SHA-1 and SHA-2).

Remark. A *digital signature* is an encrypted message digest that is appended to a document to be stored or transmitted. A message digest is obtained by using, for example, the MD5, SHA-1, or SHA-2 algorithm. The message digest is encrypted with the private key of the party that stores or transmits the message.

Data Conversion Data conversion is the conversion of one form of computer data to another. For example, conversion of a file from one version of Microsoft Word to an earlier version for the sake of those who do not have the latest version of Word installed. Data conversion testing is testing of programs or procedures that are used to convert data from existing systems for use in replacement systems. Data may be converted into an invalid format that cannot be processed by the new system if this is not performed properly; thus the data will have no value.

In addition, data may be omitted from the conversion process resulting in gaps or system errors in the new system. Inability to process backup or archive files results in the inability to restore or interrogate old data.

An acceptance criterion for data conversion measures and reports the capability of the software to convert existing application data to new formats. The following questions must be answered in specifying the data conversion acceptance criteria:

- How can we undo a conversion and roll back to the earlier database version(s) if necessary?
- How much human involvement is needed to validate the conversion results?
- How are the current data being used and how will the converted data be used?
- Will the data conversion software conduct integrity checking as well?

Backup and Recovery Backup and recovery of data are default functionalities of large, complex systems. This is because, though systems are not expected to crash, in reality, a system crash is not uncommon. The backup and recovery acceptance criteria specify the durability and recoverability levels of the software in each hardware platform. The aim of the recovery acceptance test criteria is to outline the extent to which data can be recovered after a system crash. The following questions must be answered in specifying the recoverability acceptance criteria:

- How much data can be recovered after a crash and how?
- Is the recovered data expected to be consistent?

Generally, a system cannot recover from a crash unless the data have been previously backed up. The backup process includes taking periodic snapshots of a state of the system and saving them in stable storage to be retrieved later [5]. The following questions must be answered in specifying the backup acceptance criteria:

- How frequently is the backup process initiated?
- How long does the backup process take?
- Is the backup expected to work on-line or off-line with normal operation suspended during backup?
- Does the backup process check if sufficient storage space is available to accommodate all the data?
- Is the backup process fully automated?

Competitive Edge The system must provide a distinct advantage over existing methods and competing products through innovative features. An analysis of the competitiveness of the product is provided to the buyer. This document contains a comparative study of the system with products available in the market from other vendors. A competitive analysis is conducted by the systems engineering group

of the marketing organization. The following questions need to be answered in specifying the competitive analysis report acceptance criteria:

- What are the nearest direct competitors of the product?
- What are the indirect competitors of the product?
- Who are the potential competitors?
- Is the business in the product area steady, growing, or declining?
- What can be learned from product operations or from advertisements of competitors?
- What are the strengths and weaknesses of competitors?
- How do their products differ from the product being developed?

Usability The question is: How easy it is to use the system and how easy it is to learn? The goal of usability acceptance criteria is to ensure that the system is flexible, it is easy to configure and customize the system, on-line help is available, work-around is available, and userinterface is friendly. The following questions need to be addressed in specifying the usability acceptance criteria:

- How will the system help the user in the day-to-day job?
- Will the productivity, customer satisfaction, reliability, and quality of work life of the user improve by using the system?
- Are the menus, commands, screens, and on-line help clear to a typical user?
- Are the user procedures simple, logical, and clear to the typical user?
- Is the user guide clear, easy to access, and understandable for a typical user?
- Will the methods of error and exception handling utilized by the system increase reliability and productivity?
- Are the reports generated by the system in order, consistent, and clear?
- Is the system easy to install?

Performance The desired performance characteristics of the system must be defined for the measured data to be useful. The following questions relate to specification of the performance acceptance criteria:

- What types of performance characteristics of the system need to be measured?
- What is the acceptable value for each performance parameter?
- With what external data source or system does the application interact?
- What kind of workload should be used while running the performance tests? The workload should be a representative of the likely real-world operating condition in terms of low load, average load, and peak load.

- Is it required to perform a before-and-after comparison of the performance results with the prior version of the system?

Start-Up Time The system start-up time reflects the time taken to boot up to become operational. The following questions address the start-up acceptance criteria:

- How is the start-up time defined?
- Does the start-up time include the power-on self-test of all the system hardware?
- What is the longest acceptable start-up time?

Stress The system should be capable of handling extremely high or stressful load. It is necessary to identify the system limitations and then stress the system to find the results when the system is pushed to the border and beyond. The system limitation must be identified in the acceptance criteria. The following questions must be addressed in specifying the stress acceptance criteria:

- What are the design limits of the system?
- What is the expected and acceptable behavior of the recovery mechanism?
- What test environment, close to customer deployment architecture, is needed in order to force the system to be stressed?

Reliability and Availability Software reliability is defined as the probability that the software executes without failure for a specified amount of time in a specified environment. The longer a system runs without failure, the more reliable it is. A large number of reliability models are available to predict the reliability of software. A software reliability model provides a family of growth curves that describe the decline of failure rate as defects are submitted and closed during the system testing phase. The failure rate is often calculated in terms of MTBF. A growth model can answer the following questions, which can be part of the reliability acceptance criteria:

- What is the current failure rate of the software?
- What will be the failure rate if the customer continues acceptance testing for a long time?
- How many defects are likely to be in the software?
- How much testing has to be performed to reach a particular failure rate?

The failure rate goal that is acceptable must be set separately for each level of problem severity—from *critical* to *low*. A customer may be willing to tolerate tens of low-severity issues per day but not more than one critical problem in a year.

System availability consists of proactive methods for maximizing service uptime, for minimizing the downtime, and for minimizing the time needed to

recover from an outage. Downtime is measured in terms of MTTR. The creation of a customer environment is facilitated by gathering an *operational profile* from the customer. An operational profile describes the ways the system is to be used. One can uncover several deficiencies in the system while tuning the parameters of the system; parameter tuning will improve system availability level. Customers must be willing to share the operational profile of their computing environment to improve the target availability level, which may be proprietary information.

Maintainability and Serviceability The maintainability of a system is its ability to undergo repair and evolution. One way to characterize maintainability is to measure the MTTR, which reflects the time it takes to analyze a *corrective* defect, design a modification, implement the change, test it, and distribute it. The important factors, from a customer's perspective, is the responsiveness of the service rather than the internal technical maintainability of the system. The following are useful acceptance criteria from a customer's perspective:

- The customer is the final arbiter of setting the severity of a system problem. If the customer calls a problem critical, it must be fixed immediately.
- If a system problem is assessed as critical by the customer, then staff must be assigned to work on resolving the problem immediately with utmost priority.
- If the severity of a system problem is assessed as high by the customer, then staff must be assigned to work on resolving the problem during normal business hours until it is resolved or until a work-around has been delivered as an interim solution. The staff responsible for resolving the problem must ensure that there is significant effort made toward resolving the problem. However, they may spend time on other activities as priorities dictate.
- If a system problem is assessed as low by the customer, then staff must be assigned to work on resolving the problem during normal business hours as time permits. If the problem solution involves a software change, it will normally wait until the next software release has been implemented to provide the resolution.
- All the critical- and high-severity fixes must work 100% when installed.

Serviceability is closely related to maintainability of the system, which are designed to ensure the correctness of the tools that are used to diagnose and service the system. For example, the software may need to be serviced remotely via an Internet connection. Diagnostic utilities are used to monitor the operation and the cause of any malfunction. The following questions must be addressed in specifying the serviceability acceptance criteria:

- What kind of tools will be available for servicing the system?
- How should these tools be used?

Robustness The robustness of a system is defined as its ability to recover from errors, continue to operate under worst conditions, and operate reliably for an extended period of time. The following questions must be addressed in specifying the robustness acceptance criteria:

- What are the types of errors from which the system is expected to recover?
- What are the causes, or sources, of the errors so that these can be simulated in a test environment?
- How are the errors initiated, or triggered, in the real world?
- What types of corrective and recovery actions are required for each type of error?
- What kinds of disasters can strike? What are those scenarios?
- What is an acceptable response to each of these identified scenarios?
- What is the recovery mechanism for each of the scenarios? Is it workable, understood, and accepted?
- How can disaster be simulated in order to test recovery?

Timeliness Time to market is an important aspect of any contractual agreement. The supplier must be able to deliver the system to the buyer within the time frame agreed upon. Rewards and penalties are associated with the timeliness acceptance criteria as follows:

- If coding is completed on time, the buyer will reward 5% extra money on top of the contractual agreement.
- If system-level testing is completed on time, the buyer will reward 10% extra money on top of the contractual agreement.
- For every week of delay in completing the system tests, the supplier has to pay 2% penalty on top of the contractual agreement, with a maximum of 20% penalty.

Confidentiality and Availability The confidentiality acceptance criteria reter to the requirement that the data must be protected from unauthorized disclosure and the availability acceptance criteria to the requirement that the data must be protected from a denial of service (DoS) to authorized users. Different types of possible confidentiality and availability acceptance criteria are as follows:

- No unauthorized access to the system is permitted, that is, user authentication is performed.
- Files and other data are protected from unauthorized access.
- The system is protected against virus, worm, and bot attacks.
- Tools are available for detecting attacks.
- There is support against DoS attack.
- Privacy in communication is achieved by using encryption.

- All the customer data must be stored in a secure place in accordance with the policies of customer right, such as confidentiality.

Remark. A worm is defined as a software component that is capable of, under its own means, infecting a computer system in an automated fashion. On the other hand, a virus spreads rapidly to a large number of computers. However, it cannot do so with its own capability; it spreads using the assistance of another program.

Remark. A bot is a software agent. A bot interacts with other network services intended for people as if it were a person. One typical use of bots is to gather information. Another more malicious use for bots is the coordination and operation of an automated attack on networked computers, such as a distributed DoS attack.

Compatibility and Interoperability The compatibility of a system is defined as the ability to operate in the same way across different platforms and network configurations and in the *presence* of different mixes of other applications. On the other hand, the interoperability of a system is defined as the ability to *interface* with other network elements and work correctly as expected. The major challenge is in determining the platforms, configurations, and other applications with which the system is compatible. The following questions must be addressed in specifying the compatibility and interoperability acceptance criteria:

- What are the platforms, or configurations, on which the system must operate?
- Does the system have to work exactly the same way across different configurations? If not, what are the acceptable variations?
- What are the applications that must coexist with the system?
- With what network elements must the system interoperate?

Compliance The system should comply with the relevant technical standards, such as the IEEE standards, operating system interface standards, and the IP standards. In addition, the system should comply with regulatory requirements as established by external agencies. The following questions must be addressed in specifying the acceptance criteria for compliance:

- With what technical standards should the system comply? Are there any exceptions to these standards? If yes, specify the exceptions.
- Identify the regulatory bodies that must certify the system?

Installability and Upgradability The purpose of system installability and upgradability is to ensure that the system can be correctly installed and upgraded in the customer environment. If for some reason the customer wants to uninstall or downgrade the system software, it is required to be done smoothly. Installation and upgradation of a system is planned by identifying the major milestones and

contingency steps. The system installation and upgradation process document must be available with specific steps. The acceptance criteria of system installation and upgradation are as follows:

- The document must identify the person to install the system, for example, the end user or a trained technician from the supplier side.
- Over what range of platforms, configurations, and versions of support software is the installation or upgradation process expected to work? The hardware and software requirements must be clearly explained in the document.
- Can the installation or upgradation process change the user's existing environment? If yes, the risks of this change should be clearly documented.
- The installation or upgradation process should include diagnostic and corrective steps to be used in the event of the process not progressing as expected.
- The installation or upgradation process should contain a workable uninstall, downgrade, or backoff process in case a specific installation does not proceed as expected.
- The installation or upgradation process should correctly work from all of the various delivery media, such as download via File Transfer Protocol (FTP), CD-ROM, and DVD.
- If the system includes a licensing and registration process, it should work smoothly and should be documented.
- The installation or upgradation instructions should be complete, correct, and usable.
- The installation or upgradation process should be verified during system testing.
- There should be zero defects outstanding against a system installation or upgradation process.

Scalability The scalability of a system is defined as its ability to effectively provide acceptable performance as the following quantities increase: (i) geographic area of coverage of a system, (ii) system size in terms of the number of elements in the system, (iii) number of users, and (iv) volume of workload per unit time. A system may work as expected in limited-use scenarios but may not scale up very well. The following questions must be addressed in specifying the scalability acceptance criteria:

- How many concurrent users is the system expected to handle?
- How many transactions per unit time is the system expected to process?
- How many database records is the system expected to support?
- How many elements, or objects, must be managed in live operation?
- What is the largest geographic area the system can cover?

Documentation The quality of the system user's guide must be high. The documentation acceptance criteria are formulated as follows:

- All the user documents should be reviewed and approved by the software quality assurance group for correctness, accuracy, readability, and usefulness.
- The on-line help should be reviewed and signed off by the software quality assurance group.

ESTO YA FORMARIA PARTE DEL ÚLTIMO TEMA

14.3 SELECTION OF ACCEPTANCE CRITERIA

The acceptance criteria discussed above provide a broad idea about customer needs and expectations, but those are too many and very general. The customer needs to select a subset of the quality attributes and prioritize them to suit their specific situation. Next, the customer identifies the acceptance criteria for each of the selected quality attributes. When the customer and the software vendor reach an agreement on the acceptance criteria, both parties must keep in mind that satisfaction of the acceptance criteria is a trade-off between time, cost, and quality. As Ed Yourdon opined, sometimes less than perfect is good enough [6]. Only business goals and priority can determine the degree of "less than perfect" that is acceptable to both the parties. Ultimately, the acceptance criteria must be related to the business goals of the customer's organization.

Many organizations associated with different application domains have selected and customized existing quality attributes to define quality for themselves, taking into consideration their specific business and market situation. For example, IBM used the quality attribute list CUPRIMDS—capability, usability, performance, reliability, installation, maintenance, documentation, and service—for its products [7]. Similarly, for web-based applications [8], a set of quality attributes are identified in decreasing order of priority: reliability, usability, security, availability, scalability, maintainability, and time to market. Such a prioritization scheme is often used in specific application domains. For example, usability and maintainability take precedence over performance and reliability for a word processor software. On the other hand, it might be the other way around for a real-time operating system or telecommunication software.

14.4 ACCEPTANCE TEST PLAN (PARA EL ÚLTIMO TEMA)

Planning for acceptance testing begins as soon as the acceptance criteria are known. Early development of an acceptance test plan (ATP) gives us a good picture of the final product. The purpose of an ATP is to develop a detailed outline of the process to test the system prior to making a transition to the actual business use of the system. Often, the ATP is delivered by the vendor as a contractual agreement, so that the business acceptance testing can be undertaken within the vendor's development organization to ensure that the system eventually passes the acceptance test.

In developing an ATP, emphasis is put on demonstrating that the system works according to the customer's expectation, rather than passing a set of comprehensive tests. In any case, the system is expected to have already passed a set of comprehensive tests during system-level testing. The ATP must be kept very simple because the audience of this plan may include people from diverse backgrounds, such as marketing and business managers. Some people argue that the ATP is redundant and unnecessary if a comprehensive system test plan is developed. We believe that even if a system test plan is adequate, acceptance tests usually uncover additional significant problems. Moreover, user's concerns are not addressed during system-level testing.

An ATP needs to be written and executed by the customer's special user group. The user group consists of people from different backgrounds, such as software quality assurance engineers, business associates, and customer support engineers. In addition, the acceptance test cases are executed at the user's operational environment, whereas the system-level test cases are executed in a laboratory environment. An overall test plan for acceptance testing and description of specific tests are documented in the ATP. The structure of a typical ATP is outlined in Table 14.1.

The introduction section of the ATP describes the structure of the test plan and what we intend to accomplish with this test plan. This section typically includes (i) test project name, (ii) revision history, (iii) terminology and definitions, (iv) names of the approvers and the date of approval, (v) an overview of the plan, and (vi) references.

For each quality category from the acceptance criteria signed-off document two subsections are created: operational environment and test case specification. The operational environment deals with discussion on site preparation for the execution of the acceptance test cases. Test cases are specified for each acceptance criteria within the quality category.

An outline of the timeline of execution of acceptance tests is provided in the schedule section of the ATP. Acceptance test execution is not intended to be exhaustive, and therefore it does not continue for long. The acceptance test

TABLE 14.1 Outline of ATP

1. Introduction
2. Acceptance test category. For each category of acceptance criteria:
  (a) Operational environment
  (b) Test case specification
    (i) Test case ID number
    (ii) Test title
    (iii) Test objective
    (iv) Test procedure
3. Schedule
4. Human resources

may take up to six weeks for a large system. The point here is that comprehensive acceptance testing, to the same extent and depth as targeted by system-level testing, is not required to demonstrate that the acceptance criteria are satisfied by the system.

The human resources section of the ATP deals with (i) the identification of the acceptance testers that form the client organization and (ii) their specific roles in the execution of acceptance test cases. The section includes acceptance test site preparation, overseeing installation of new hardware, upgrading the software, and setting up of the networks. These are the people who are knowledgeable in the operational environment and business operations. In addition, the human resources requirement from the supplier organization during the acceptance testing is included in this section. These engineers are usually from the supplier's system test group, who participated in testing the system.

The ATP is reviewed and approved by the relevant groups, such as the marketing, customer support, and software quality assurance groups. It can be shared with the system supplier organization.

14.5 ACCEPTANCE TEST EXECUTION

The acceptance test cases are divided into two subgroups. The first subgroup consists of basic test cases, and the second consists of test cases that are more complex to execute. The acceptance tests are executed in two phases. In the first phase, the test cases from the basic test group are executed. If the test results are satisfactory, then the second phase, in which the complex test cases are executed, is taken up. In addition to the basic test cases, a subset of the system-level test cases are executed by the acceptance test engineers to independently confirm the test results. Obviously, a key question is: Which subset of the system-level test cases are selected? It is recommended to randomly select 5–10 test cases from each test category. If a very large fraction, say more than 0.95, of the basic test cases pass, then the second phase can proceed. It may be counterproductive to carry out the execution of the more complex tests if a significant fraction of the basic tests fail.

Acceptance test execution is an important activity performed by the customer with much support from the developers. The activity includes the following detailed actions:

- The developers train the customer on the usage of the system.
- The developers and the customer coordinate the fixing of any problem discovered during acceptance testing.
- The developers and the customer resolve the issues arising out of any acceptance criteria discrepancy.

System-level test personnel from the development organization travel to the customer location where the acceptance tests are to be conducted. They assist the

TABLE 14.2 ACC Document Information

1. ACC number	A unique number
2. Acceptance criteria affected	Existing acceptance criteria
3. Problem/issue description	Brief description of issue
4. Description of change required	Description of changes needed to be done to original acceptance criterion
5. Secondary technical impacts	Description of impact it will have on system
6. Customer impacts	Impact it will have on end user
7. Change recommended by	Name of acceptance test engineer(s)
8. Change approved by	Name of approver(s) from both parties

customer in preparing a test site and train the acceptance test engineers on how to use the system. They provide the earlier system-level test results to the customer's test engineers in order to make informal decisions about the direction and focus of the acceptance testing effort. In addition, the on-site system test engineers answer the customer's questions about the system and assist the acceptance test engineers in executing the acceptance tests.

Any defect encountered during acceptance testing are reported to the software development organization through the on-site system test engineers. The defects are submitted through the defect tracking system. The software build is retested by the supplier and a satisfactory software image is made available to the customer for continuation of acceptance testing when the defects are fixed. The failed tests are repeated after the system is upgraded with a new software image. An agreement must be reached between the on-site system test engineers and the acceptance test engineers when to accept a new software image for acceptance testing. The number of times the system can be upgraded to a new software image during acceptance testing is negotiated between the customer and the supplier. Multiple failures of a system during acceptance testing are an indication of poor system quality.

It is possible that an acceptance test engineer may encounter issues related to acceptance criteria during the execution of acceptance test cases. The system may not provide services to the users as described in the acceptance criteria. Any deviation from the acceptance criteria discovered at this stage may not be fixed immediately. The acceptance test engineer may create an acceptance criteria change (ACC) document to communicate the deficiency in the acceptance criteria to the supplier. A representative format of an ACC document is shown in Table 14.2. An ACC report is generally given to the supplier's marketing department through the on-site system test engineers.

14.6 ACCEPTANCE TEST REPORT NO

Acceptance test activities are designed to reach one of three conclusions: Accept the system as delivered, accept the system after the requested modifications have

been made, or do not accept the system. Usually some useful intermediate decisions are made before making the final decision:

- A decision is made about the continuation of acceptance testing if the results of the first phase of acceptance testing are not promising. One may recall that the basic tests are executed in the first phase.
- If the test results are unsatisfactory, changes will be made to the system before acceptance testing can proceed to the next phase.

The intermediate decisions are made based on evaluation of the results of the first phase of testing. Moreover, during the execution of acceptance tests, the status of testing is reviewed at the end of every working day by the leader of the acceptance test team, on-site system test engineers, and project managers of the customer and the supplier. The acceptance team prepares a test report which forms the basis of discussion at the review meeting before they meet for a review. A template of the test report is given in Table 14.3.

The test report is reviewed on a daily basis to understand the status and progress of acceptance testing. If serious problems are encountered during acceptance testing, the project manager flags the issues to the senior management.

At the end of the first and the second phases of acceptance testing an acceptance test report is generated by the test team leader. A template for a test report is outlined in Table 14.4. Most of the information from the test status report can be used in the acceptance test summary report.

The report identifier uniquely identifies the report. It is used to keep track of the document under version control.

The summary section summarizes what acceptance testing activities took place, including the test phases, releases of the software used, and the test environment. This section normally includes references to the ATP, acceptance criteria, and requirements specification.

The variances section describes any difference between the testing that was planned and the actual testing carried out. It provides an insight into a process for improving acceptance test planning in the future.

TABLE 14.3 Structure of Acceptance Test Status Report

1. Date	Acceptance report date
2. Test case execution status	Number of test cases executed today
	Number of test cases passing
	Number of test cases failing
3. Defect identifier	Submitted defect number
	Brief description of issue
4. ACC number(s)	Acceptance criteria change document number(s), if any
5. Cumulative test execution status	Total number of test cases executed
	Total number of test cases passing
	Total number of test cases failing
	Total number of test cases not executed yet

TABLE 14.4 Structure of Acceptance Test Summary Report

1. Report identifier
2. Summary
3. Variances
4. Summary of results
5. Evaluation
6. Recommendations
7. Summary of activities
8. Approval

In the summary of results section of the document test results are summarized. The section gives the total number of test cases executed, the number of passing test cases, and the number of failing test cases; identifies all the defects; and summarizes the acceptance criteria to be changed.

The evaluation section provides an overall assessment of each category of the quality attributes identified in the acceptance criteria document, including their limitations. This evaluation is based on the test results from each category of the test plan. The deviations of the acceptance criteria that are captured in the ACC during the acceptance testing are discussed.

The recommendations section includes the acceptance test team's overall recommendation: (i) unconditionally accept the system, (ii) accept the system subject to certain conditions being met, or (iii) reject the system. However, the ultimate decision is made by the business experts of the supplier and the buyer organization.

The summary of activities section summarizes the testing activities and the major events. This section includes information about the resources consumed by the various activities. For example, the total manpower involved in and the time spent for each of the major testing activities are given. This section is useful to management for accurately estimating future acceptance testing efforts.

Finally, the names and titles of all the people that will approve this report are listed in the approvals section. Ideally, the approvers of this report should be the same people who approved the corresponding ATP because the summary report describes all the activities outlined in the ATP. If some of the reviewers have minor disagreements, they may note their views before signing off on the document.

14.7 ACCEPTANCE TESTING IN eXtreme PROGRAMMING

NO. HABLAREMOS DE ESTO EN EL ÚLTIMO TEMA

In the XP[9] framework user stories are used as acceptance criteria. The user stories are used to derive time estimates for each development iteration in release planning (time-to-market acceptance criteria) and acceptance tests. The user stories are written by the customer as things that the system needs to do for them. The

stories are usually about two to three sentences of text written using the customer's terminology. Several acceptance tests are created to verify that the user story has been correctly implemented. Acceptance tests are specified in a format that is clear enough that the customer can understand and specific enough that it can be executed.

The customer is responsible for verifying the correctness of the acceptance tests and reviewing the test results [10]. The acceptance test results are reviewed by the customer to decide what failed tests are of highest priority that must pass during the next iteration. A story is incomplete until it passes its associated acceptance tests.

Acceptance tests are executed by an acceptance test group, which is a part of the development team. Ideally, acceptance tests should be automated using either the unit testing framework or a separate acceptance testing framework before coding. Acceptance test engineers and customers can run the tests multiple times per day as a regression acceptance test suite after the acceptance tests are automated. An automated acceptance test suite does not lose its value even after the customer has approved the successful implementation of the user story in a development iteration. The acceptance tests take on the role of regression tests to ensure that subsequent changes to the system do not affect the unaltered functionality.

14.8 SUMMARY

This chapter began with an introduction to two types of acceptance testing: user acceptance testing and business acceptance testing. Next, the chapter described acceptance criteria in terms of quality attributes. Formulation of acceptances criteria is governed by the business goals of the customer's organization.

We presented an outline of an acceptance test plan and described in detail how to create such a plan. Emphasis must be put on the notion that the system works according to the customer's expectations in developing an acceptance test plan, rather than just passing comprehensive testing. Less emphasis is put on a system passing a comprehensive set of tests because rigorous testing is assumed to have already occurred during the system testing phase.

Next, we discussed the execution of acceptance tests, which is an important activity performed by the customer with much needed support from the developers. Three major activities were identified and discussed: (i) providing training to the customer's test engineers, (ii) fixing problems during acceptance testing, and (iii) resolving issues concerning any discrepancy related to acceptance criteria. After that, we described the generation of an acceptance test report, which must be completed at the end of acceptance testing.

Finally, we explained how user stories are used in XP as acceptance criteria and acceptance test cases are created. These tests are reviewed, automated, and executed multiple times per day as a regression acceptance test suite in the presence of on-site customers.

LITERATURE REVIEW

A thorough treatment of software quality standard can be found in ISO/IEC 9126-1:2001, "Software Engineering—Product Quality—Part-I: Quality Model." The standards group has recommended a hierarchical framework for software quality characteristics and subcharacteristics. Six top-level software quality characteristics are *functionality, reliability, usability, efficiency, maintainability*, and *portability*.

An excellent taxonomy in the field of the *dependability* quality attribute is presented in the article by Algirdas Avižienis, Jean-Claude Laprie, Brian Randell, and Carl Landwehr, "Basic Concepts and Taxonomy of Dependable and Secure Computing," *IEEE Transactions on Dependable and Secure Computing*, Vol. 1, No. 1, January–March 2004, pp. 11–33. In this article, the relationship between dependability and *security* quality attributes is discussed in great detail. Dependability is defined as an integrating concept that encompasses the following attributes: availability, reliability, safety, integrity, and maintainability. On the other hand, security brings in concerns for confidentiality in addition to availability and integrity. The article addresses the concept of threats to dependability and security (faults, errors, failures), their attributes, and the means for their achievement, such as fault prevention, fault tolerance, fault removal, and fault forecasting.

An interesting view of software quality characteristics is presented in C. K. Prahalad and M. S. Krishnan, "The New Meaning of Quality in the Information Age," *Harvard Business Review*, September 1999, pp. 109–118. The authors provide a framework for measuring the performance of software in a company's information technology portfolio based on three basic attributes: conformance, adaptability, and innovation. To judge software quality properly, the authors argue that managers must measure applications against all the three attributes.

The students are encouraged to read the article by P. Hsia, D. Kung, and C. Sell, "Software Requirements and Acceptance Testing," *Annals of Software Engineering*, Vol. 3, No. 0, January 1997, pp. 291–317. The authors present a systematic approach to scenario analysis and its application to acceptance testing. Based on the model, different acceptance test criteria are defined, and various types of acceptance testing are discussed.

The book entitled *Testing Extreme Programming* (L. Crispin and T. House, Addison-Wesley, Boston, MA, 2003) gives an excellent exposition of unit and acceptance testing in XP. The book is divided into three parts: the XP tester role, test drive through an XP project, and road hazard survival kit. The book lists several excellent books on the subject in its bibliography.

REFERENCES

1. Institute of Electrical and Electronics Engineers (IEEE). *IEEE Standard Glossary of Software Engineering Terminology*. IEEE standard 610.12-1990. IEEE, New York, 1990.
2. B. Kitchenham and S. L. Pfleeger. Software Quality: The Elusive Target. *IEEE Software*, January 1996, pp. 12–21.