**Chapter 12: Construction  
Testbank**

**Multiple Choices**

1. When information systems projects fail, the primary reason has traditionally been \_\_\_\_\_.

a. improperly trained programmers

b. inadequate planning

c. poor analysis, design, installation, or project management

d. poor programming

e. shortened testing periods

Ans: c

1. \_\_\_\_\_ is the development of all parts of the new system including the software, documentation, and new operating procedures.

a. Construction

b. Design

c. Documentation

d. Management

e. Testing

Ans: a

1. The cost of fixing one major bug after the system is installed can easily exceed the \_\_\_\_\_.

a. annual salary of a programmer

b. cost of one week’s work as a programmer

c. cost of two weeks’ work as a programmer

d. entire cost of the programming project

e. value of the entire systems team

Ans: a

1. When assigning programmers to a programming team, the best size of the team is \_\_\_\_\_.

a. five

b. six

c. the largest number of people possible

d. the smallest number of people possible

e. twenty-two

Ans: d

1. Regular meetings of a systems project team \_\_\_\_\_.

a. encourage communication

b. is not a simple approach

c. reduce communication

d. reduce discussion

e. should not be held unless necessary

Ans: a

1. When programmers or project teams set up three areas on different disk drives in which the programmers can work, these areas are for \_\_\_\_\_.

a. development, goal setting, and testing

b. development, testing, and contracting

c. development, testing, and production

d. testing, filing, and printing

e. testing, printing, and program logging

Ans: c

1. A classic mistake made during the implementation phase is to \_\_\_\_\_.

a. create a risk assessment

b. maintain control over the code

c. plan for the use of state-of-the-art technology

d. spend too much time in testing

e. use low cost personnel

Ans: e

1. Which of the following statements is true about testing object-oriented systems?

a. testing can be minimized since we are reusing parts of systems that have been tested before

b. testing should be designed to prove there are no errors in the system

c. testing is more important than it was in systems developed before object-oriented development

d. encapsulation makes testing easier

e. most testing techniques have been developed to support object-oriented development

Ans: c

1. \_\_\_\_\_ allow hiding everything in a system behind a visible interface, making testing difficult.

a. Encapsulation and information hiding

b. Polymorphism and dynamic binding

c. Inheritance

d. Reuse

e. Object-oriented development processes

Ans: a

1. \_\_\_\_\_ makes it impossible to know which implementation is going to be executed until the system actually does it.

a. Information hiding

b. Dynamic binding

c. Inheritance

d. Reuse

e. Encapsulation

Ans: b

1. The number of errors found will be highest in the \_\_\_\_\_ stage of testing.

a. unit test

b. integration test

c. system test

d. acceptance test (alpha)

e. acceptance test (beta)

Ans: b

1. A(n) \_\_\_\_\_ defines a series of tests that will be conducted to identify errors in programming.

a. object test

b. test script

c. stud test

d. test case

e. test plan

Ans: e

1. Mary has written a message into her program to notify her when a particular class has been reached within the program. This message is commonly called a(n) \_\_\_\_\_.

a. change item

b. error message

c. navigator

d. stub

e. unit

Ans: d

1. \_\_\_\_\_ is the type of test performed to assess whether a set of classes that must work together do so without an error.

a. Acceptance testing

b. Beta testing

c. Integration testing

d. System testing

e. Unit testing

Ans: c

1. Which approach to system testing checks to see if the new system can handle high volumes of transactions?

a. requirements testing

b. usability testing

c. security testing

d. performance testing

e. documentation testing

Ans: d

1. \_\_\_\_\_ is the test that is performed to ensure that all classes work together without error and meet the business requirements for usability, security, and performance.

a. Acceptance testing

b. Beta testing

c. Integration testing

d. System testing

e. Unit testing

Ans: d

1. \_\_\_\_\_ documentation is designed to help programmers and systems analysts understand the application software and maintain it after the system is installed.

a. Acceptance

b. Integration

c. Navigation

d. System

e. User

Ans: d

1. The type of documentation designed to be used when the user needs to learn how to perform a specific function is known as \_\_\_\_\_.

a. procedural cohesion

b. procedures manuals

c. reference documents

d. sequential cohesion

e. tutorials

Ans: c

1. A(n) \_\_\_\_\_ is a navigation control for on-line documentation that provides access into topics using important keyword.

a. agent search

b. index

c. table of contents

d. text search

e. web-like links

Ans: b

1. Topics in the text documentation should be written from the viewpoint of \_\_\_\_\_\_.

a. the programmers

b. the users

c. the sponsor

d. what the system can do

e. the objects

Ans: b

1. In developing documentation use \_\_\_\_\_\_.

a. the active voice

b. e-prime style

c. consistent terms

d. parallel grammatical structures

e. all of these

Ans: e

1. In developing documentation use \_\_\_\_\_\_\_\_\_\_.

a. formal language

b. simple language

c. technical language

d. all of these

e. none of these

Ans: b

1. Hall identified several cultural differences, and cultural issues add a new wrinkle in the management of developing a successful information system. From an information systems development perspective, \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ could influence the ability of a team member to see (or not see) potential creative solutions that are out of the box or affect a team member’s ability (or inability) to understand the entire problem under consideration.

a. time

b. context

c. color

d. language

e. geography

Ans: b

1. Hofstede identified 5 dimensions on cultural differences. Which one in the following list is NOT the dimension Hofstede identified?

a. individualism and collectivism

b. time

c. power distance

d. religion

e. masculinity versus femininity

Ans: d

1. From an information systems development perspective, the common language today tends to be \_\_\_\_\_\_\_\_\_\_\_\_, Java, SQL, C++, Objective-C, and Visual Basic.

a. English

b. Formal language like Finite State Machines

c. technical language

d. UML

e. Graphical language

Ans: d

1. Security testing involves three primary areas: \_\_\_\_\_\_\_\_, authorization, and virus control.

a. encoding/decoding

b. monitoring

c. authentication

d. validation

e. verification

Ans: c

1. Which one in the following is NOT in the state of the system testing?

a. Requirements Testing

b. Usability Testing

c. Security Testing

d. Performance Testing

e. System Interface Testing

Ans: e

1. In addition to testing the functional requirements across all layers, testing must also address all nonfunctional requirements. Which of the following is not one of those nonfunctional requirements?

a. Operational

b. Budgetary

c. Performance

d. Security

e. Political requirements

Ans: b

**True/False**

1. Programming can be the greatest single component of any information systems development project in terms of time and cost.

Ans: True

1. Implementation is the development of all parts of the system, including the software development, documentation creation, and new operating procedures generation.

Ans: False

1. The programming process of a new systems project is considered completed prior to testing.

Ans: False

1. The larger the number of programmers assigned to a programming team, the shorter the time will be for programming to be completed.

Ans: False

1. A test plan is often a short document (1 to 2 pages) that has one to two sentences about each test to be performed during the testing part of the implementation phase.

Ans: False

1. Integration tests focus on one unit, program, or class that performs a specific function.

Ans: False

1. System tests are conducted by the systems analysts to ensure that all classes work together without error.

Ans: True

1. Three common types of documentation are reference documents, help systems, and procedure manuals.

Ans: False

1. Three recommended guidelines for writing documentation are to use an active voice, use consistent terms, and use parallel grammatical structures.

Ans: True

1. Documentation should be written in long paragraphs so that users get large amounts of detail quickly.

Ans: False

1. If you use state-of-the-art technology, you can significantly decrease the project’s time and cost estimates because the newer technologies are much faster and more efficient than older ones.

Ans: False

1. If cost is a critical issue, you should assign the best, most expensive personnel to the implementation phase; never assign entry-level personnel in an attempt to save costs.

Ans: True

1. The number one reason for project failure during implementation is testing in which the programmers and analysts test the system without formal test plans.

Ans: True

1. On large projects, programmers must coordinate changes to the program source code, and the best way to avoid mistakes is to use a source code library, which requires programmers to check out programs and prohibits others from working on them at the same time.

Ans: True

1. The four approaches to unit testing are user interface testing, use scenario testing, data flow testing, and system interface testing; most projects use all four approaches.

Ans: False

1. Acceptance testing is done in two stages: alpha testing, where users test the system using made-up data, and beta testing, where users begin to use the system with real data but are carefully monitored for errors.

Ans: True

1. System documentation is intended to help programmers and systems analysts understand the application software and enable them to build it or maintain it after the system is installed.

Ans: True

1. Paper-based user documentation is simpler to use because it is more familiar to users, is easier to flip through to gain a general understanding of its organization and topics, and can be used far away from the computer itself.

Ans: True

1. One of the most common causes for schedule problems is scope creep

Ans: True

1. If cost is a critical issue, a reasonable solution is to assign lower cost programmers to the project.

Ans: False

1. A classic mistake made during the implementation phase is to coordinate changes to the source code, since that increases the development time.

Ans: False

1. Unit test is the type of test performed to ensure that a class is performing the function as it is defined in the program specifications.

Ans: True

1. Interaction testing is one type of integration testing. In interaction testing, the entire system starts as a set of stubs, and each class is added in turn and tested.

Ans: True

1. Beta testing is part of performance testing.

Ans: False

1. System documentation consists of user’s manuals, training manuals, and on-line help systems.

Ans: False

1. Performance testing (examining the ability of the system to perform under high loads) is an example of integration testing

Ans: False

1. Time required to develop and test user documentation should be built into the project plan.

Ans: True

1. Unit testing includes two kinds of tests: white box testing and black box testing.

Ans: True

1. A reference document is designed so the user can learn how to perform a specific computer application function.

Ans: True

1. A reference document describes how to perform a business task that typically requires several computer functions or steps.

Ans: False

1. A tutorial will teach people how to use a major component of the system by providing a step-by-step or sequence of entries.

Ans: True

1. The type of documentation designed to describe how to perform business tasks is known as procedures manuals.

Ans: True

1. The general structure used in most on-line documentation for documents, procedures manuals, or tutorials is called documentation items.

Ans: False

1. The type of navigation control for topics that organizes the information into a logical form (as though the users were to read the material from start to finish) is known as table of contents.

Ans: True

1. An index is a navigation control for on-line documentation that provides the ability to search through the topics with user generated text.

Ans: False

1. The guideline for crafting documentation that creates a more readable text by putting the subject at the start of the sentence, the verb in the middle, and the object at the end of the sentence is “use the active voice”.

Ans: True

1. The guideline for crafting documentation that when followed permits the user to quickly scan text to find the information needed is “use e-prime style”.

Ans: False

1. Changing the headings from “Calling the Help Desk”, “How to Dial the Phone,” and “Provide Important Details” to “Calling the Help Desk”, “Dialing the Phone”, and “Providing Important Details” is an example of implementing the “use parallel grammatical structures” guideline for crafting documentation.

Ans: True

1. The change from “*Select the text you want to copy. Pressing the copy button will copy the marked text to the new location*.” to “*Select the text you want to copy. Pressing the copy button will copy the selected text to the new location*.” is an application of the guideline “use consistent terms” for crafting documentation?

Ans: True

1. Omitting all forms of the verb “to be” from documentation to create a more active writing style follows the documentation guideline “use active voice.”

Ans: False

1. Terms for the index and search engine can come from technical terms the analyst used in creating the system.

Ans: False

1. When writing documentation, all commands contain two parts, an index and an action.

Ans: False

1. The number of errors found will be lowest in the unit testing stage of testing.

Ans: False

1. The goal of acceptance testing is to confirm that the system is complete, meets the business needs that prompted the system to be developed, and is acceptable to the users.

Ans: True

1. System interface testing is performed at the system testing state.

Ans: False

1. White-Box testing is used when complexity of the system under testing is high.

Ans: True

1. One has to execute the system under test to know which implementation to be used when dynamic biding is implemented in the system.

Ans: True

1. Testing Object-oriented system is same as difficult as testing non-object-oriented systems.

Ans: False

1. Given the context dimension identified by Hall on cultural difference, the level of detail in direction could be varied between cultures.

Ans: True

1. In *polychromic* time cultures, deadlines are critical. This is probably why *timeboxing* has been relatively successful as a method to control projects.

Ans: False

1. Understanding cultural issues is extremely important to successfully manage international/multicultural development teams..

Ans: True

1. Performance testing typically falls into two categories: stress tests and speed tests.

Ans: False

1. Performance testing typically falls into two categories: stress tests and volume tests.

Ans: True

1. The purpose of volume tests is to push the implementation so that it may break when there is a large amount of data required to answer a user request.

Ans: True

1. Comparing managing programmers in a feminine culture than in a masculine culture, it is more important to ensure that the workplace is a supportive, non-competitive, and nurturing environment in a feminine culture.

Ans: True

1. In addition to testing the functional requirements across all layers, testing must also address all nonfunctional requirements.

Ans: True

1. In testing, the “happy path” are those activities and abilities that keep the users satisfied.

Ans: False

1. Testing should be limited to the functional requirements.

Ans: False

1. Testing should end once a system is deployed.

Ans: False

1. Testing is of limited usefulness when it comes to security threats because those threats vary so widely.

Ans: False

**Short Answer**

1. During the construction phase of the SDLC the project manager must coordinate the programming activities performed by the programmers. Identify three coordination activities and provide the advantages of each.

Ans: Three coordination activities that are performed by the project manager are hold regular weekly meetings, create and follow standards, and put mechanism in place to keep the programming effort well organized.

Regular weekly meetings provide an opportunity for the project team to discuss any changes or issues to the system that have arisen during the past week. Regular meetings encourage widespread communication and discussion of issues before they become problems.

The creation of standards, ranging from simple rules to the completion of formal forms, assists the project team in the completion of the project. Historically, projects have been completed faster when a team completed standards for task coordination.

Mechanisms can be put in place to keep programmers well-organized. Three areas can be established for programmer to work: a development area, a testing area, and a production area. A program is written in the development area. When it is completed it is sent to the testing area. If the program passes testing, it is placed in production. If a program does not pass testing it is sent back to development. This three work area process helps manage the “change control” process.

1. What are the two types of acceptance tests and when are they used? What is the difference between the two types of tests?

Ans: The two types of acceptance tests are alpha testing and beta testing. Alpha testing is for normal acceptance testing and is conducted by the users to ensure that they accept the system. Beta testing is performed for very important systems using real data. Alpha testing often repeats previous tests but it is conducted by the users themselves. During Beta testing the users, using real data, closely monitor the system for errors or useful improvements.

1. Identify three types of user documentation. Contrast when each form of documentation should be used.

Ans: There are three types of user documentation: reference documents or help systems, procedure manuals, and tutorials.

Reference documents or help systems are designed to be used when the user needs to learn how to perform a specific function, such as updating a field or adding a new record. Since users may have already attempted to perform the function prior to reading the reference documentation, it must be particularly clear and brief.

Procedure manuals describe how to perform a business task such as printing a monthly report or taking a customer order. Each item in the procedure manual typically guides the user through a task that requires several functions. Procedure manual entries are normally much longer than reference document entries.

Tutorials teach people how to use major components of the system, such as the operating system or the accounts receivable system. Tutorial entries are much longer than procedure manual entries. Tutorials are typically designed to be read in sequence whereas reference documents and procedure manuals are designed to be read individually.

1. Explain the distinction between unit testing and integration testing.

Ans: Unit testing focuses on one unit of the system, a class, to verify that the unit performs as defined in the program specification. Integration testing focuses on a set of classes that must work together. This testing looks at the flow of control between classes and on the data that is exchanged.

1. Sara has been assigned by the project manager to serve as the testing manager for the soon-to-be-completed systems project. The project manager has made it clear that this systems project is normal in every way and they are to design a plan that would be similar to the last. They have written a normal testing plan that is very similar to the testing plan followed on the last three projects. What are the four stages of tests Sara should include in their testing plan and what is the ideal type of test to be performed for each stage.

Ans: The four stages of tests are unit testing, integration testing, system testing, and acceptance testing. The project manager has stressed a “normal” testing procedure. There is a normal test that should be performed for each stage of testing.

During unit testing the normal type of test is black-box testing. During this test the tester focuses on whether the unit meets the requirements stated in the program specifications.

The normal test for integration testing is user-interface testing. This test is done by moving through each and every menu item in the interface by either a top-down or bottom-up order. Each interface function is tested.

Requirements testing is the normal test for system testing. This test ensures that changes made as a result of integration testing did not create new errors. During requirements testing the users often pretend to be uninformed users and perform improper actions to ensure the system is immune to invalid actions.

The final testing stage is acceptance testing. Alpha testing is performed by having the users repeat previous tests performed by systems analysts to ensure they accept the system.

1. Explain the purpose of integration testing. Explain each of the four approaches that are commonly used in integration testing.

Ans: Integration testing assesses whether or not a set of classes that must work together do so without error. The four approaches are: user interface testing, use-case testing, interaction testing, and system interface testing.

In user interface testing, each interface function is tested by testing each and every menu item in the interface either a top-down or a bottom-up manner.

In use-case testing, each use case is tested by moving through each use case to ensure they work correctly. This testing is usually combined with user interface testing because it does not test all of the interfaces.

Interaction testing tests each process in a step-by-step fashion. The entire system begins as a set of stubs. Each class is added in turn and the results of the class compared to the correct result from the test data; when a class passes, the next class is added and the test rerun. This is done for each package. Once each package has passed all tests, then the process repeats integrating the packages.

System interface testing tests the exchange of data with other systems. Because data transfers between systems area often automated and not monitored directly by the users it is critical to design tests to ensure they are being done correctly.

1. Discuss the tests performed during system testing.

Ans: There are five different tests conducted during system testing: requirements testing, usability testing, security testing, performance testing, and documentation testing. Requirements testing tests to whether original business requirements are met in the new system. Usability testing tests how convenient the system is to use. Security testing is concerned with disaster recovery and blocking unauthorized access to the system. Performance testing examines the ability of the system to perform under high data loads. Finally, documentation testing assures the accuracy of the documentation for the system.

1. Discuss the reasons why on-line documentation is expected to be the dominant form of documentation for the future.

Ans: There are several strengths of on-line documentation compared to paper-based documentation. The search capability should enable users to find relevant information faster on-line compared to searching through paper. Second, it is possible to strengthen user understanding by presenting the same information in multiple formats. Third, by being part of the computer, the user can interact with the documentation in many new ways (e.g., the 'show-me' guidance in many systems). Fourth, the cost of distributing on-line documentation is much lower than paper documentation.

1. Explain the five general types of navigation controls for user documentation topics.

Ans: The documentation developers can choose from five general types of documentation navigation controls. Tables of contents organize the topics logically, but assume that the user will read the documentation through from start to finish. An index provides access to the topics by listing the important terms and keywords alphabetically. Text search permits the user to search through the topics for any text. Intelligent agents can be used to help in the search process. Finally, hyperlinks that connect topics logically permit the user to navigate among topics as needed.

1. Why is testing so important in object-oriented systems when we are using patterns, frameworks, class libraries, and components that have all be tested before?

Ans: Because of encapsulation (and information hiding), polymorphism (and dynamic binding), inheritance, and reuse, testing is much more difficult and critical. The volume of products from the OO development process also increases the importance of testing in OO development.

1. What techniques are used by project managers during the construction phase for managing the project schedule?

Ans: Project managers need to revise the time estimates as the construction step proceeds. If a program modules takes longer than expected, the expected completion date should be revised to take this into account. Scope creep may occur if requirements are added to the project after the design phase is complete. Any proposed change to the requirements during the construction phase must undergo a cost-benefit analysis before approval. Another common cause is the unnoticed day-by-day slippage in the schedule. A project manager must watch these minor slippages carefully and update the schedule accordingly. Project managers also need to revise and update risk assessment of the project as the construction phase progresses.

1. Discuss four classic implementation mistakes, and how project teams can avoid these mistakes.

Ans: The first classic mistake is research-oriented development, by which the project team attempts to use the latest new technology, which is not well understood and/or documented. To avoid this mistake, the project manager needs to significantly increase the project time and cost estimates when state-of-the-art technology that is used. A second mistake is to use low-cost personnel. If cost is a critical issue, assign the best, most expensive personnel to a project. Assigning entry-level persons to handle complex tasks is expected to cost the project in other ways. The third classic mistake is not having a source code control. Source code developed in a large programming team needs to be controlled so that programmers accidentally do not undo each other’s work. Using source control where each programmer checks out the modules he/she needs to work on, and checks them back in when they complete changes is required in moderate to large programming teams. The fourth classic mistake is inadequate testing. Ad hoc testing is the number one reason for project failures during the implementation phase. To avoid this, the project manager should allocate sufficient time in the project plan for formal testing.

1. Explain the distinction between black box testing and white box testing.

Ans: Black-box and white-box testing are different types unit testing. In black-box testing the tester treats the class as a black-box and focuses on whether the class meets the requirements stated in the specifications. In white-box testing, the tester looks inside the class and reviews the code and tests major elements of the class. White-box testing is useful for uncovering errors or incorrect assumptions that are not immediately obvious to someone who is doing black-box testing. Black-box testing is done for normal unit testing, while white-box testing is used when the complexity of the class is high.

1. Explain the distinction between user documentation and system documentation.

Ans: System documentation is intended to help programmers and systems analysts understand the application software and enable to them to extend it or maintain it after the system is installed. System documentation is largely a byproduct of the systems analysis and design process and is created as the project unfolds. In most object-oriented development environments, it is possible to automate the creation of detailed documentation for classes and methods. User documentation, such as the user’s manuals, training manuals, and online help systems, is designed to help the user operate the system. Three fundamental types of user documentation are reference documents, procedure manuals and tutorials. Reference documents (or help systems) are useful when the user needs to learn how to perform a specific function. Procedure manuals describe how to perform business tasks. Tutorials teach users on how to use major components of the system.

1. Explain four sources for identifying navigation terms.

Ans: As you prepare documentation topics, you also begin to identify terms that will be used to help user find topics. Table of contents is often the most straightforward, because it is developed from the logical structure of the documentation topics. However, navigation terms for the index and search engines can come from four distinct sources. The first source is the set of the commands in the user interface (e.g. open file, modify customer, and print all open orders). It is important to develop the index for both the action and the object parts of the command. The second source is the set of major concepts in the system. These are often found in the use cases and classes. A third source is the set of business tasks a user performs, such as making an appointment or entering a customer order. A fourth source is the set of synonyms for the navigation terms identified by the previous three sources. For example, users may think of “exit” as “quit” and “delete” as “erase”. Synonyms help the users in searching by making the documentation more useful to them.

1. When offshoring development, how could differences in Hall’s context dimension of culture affect the contribution of a team member to the successful development of an information system? What about Hall’s time or speed of messages dimensions?

Ans: From an information systems development perspective, *context* could influence the ability of a team member to see (or not see) potential creative solutions that are out of the box or affect a team member’s ability (or inability) to understand the entire problem under consideration. Furthermore, given this dimension, the level of detail in direction could be varied between cultures. When managing programmers in a multicultural setting, Hall’s *time* dimension must also be considered. In *monochromic time* cultures, deadlines are critical. This is probably why *timeboxing* has been relatively successful as a method to control projects (see Chapter 2). However, in a *polychromic time* culture, a *deadline* is nothing more than a suggestion. Obviously, when managing programmers, understanding how the culture considers time is very important to have both a successful product delivery and a successful development process. Hall’s *speed of messages* and context dimensions could also affect the manner in which this could be addressed. Depending on the culture, too much detail could be insulting, but attempting to put this issue in to a contextual frame that is culturally sensitive is difficult.

1. What are Hofstede’s five dimensions of cultural differences? How could differences in them influence the effectiveness of an information systems development team?

Ans: Hofstede’s *individualism* and *collectivism* dimension partially explains the results regarding plagiarism and cheating described above. Given the importance that intellectual property plays in IT, this potentially could be a real problem when offshoring development to a collectivist culture.

Hofstede’s other previously mentioned dimensions are *power distance*, *uncertainty avoidance*, and *masculinity versus femininity*. Managing programmers in a culture with a high power distance value will be different than with a culture with a low power distance. For example, in the US, programmers see themselves as equals to their managers. In fact, in some firms, the president of the firm will be “coding” solutions alongside of a brand new hire. This somewhat explains the growing popularity of Agile methods (see Chapter 1). In comparison, the president of the firm would never stoop down to perform the same tasks as a new hire. It would be insulting to the president and embarrassing to the new hire. With regards to uncertainty avoidance, the choice of systems development approach could be affected. In a culture that prefers everything to be neat and ordered, a system development methodology that is very rule-driven would be beneficial. Furthermore, development team member professional certification and team/firm ISO or CMMI certifications would lend credibility to the team, whereas, in a culture that willingly takes on risk, certifications may not increase the perceived standing of the development team. When managing programmers in a masculine culture, it is critical to provide recognition to the top-performing members of the development team and also to recognize the top-performing teams. On the other hand, when considering a feminine culture, it is more important to ensure that the workplace is a supportive, non-competitive, and nurturing environment.

Finally, Hofstede identified a fifth dimension, *long-* versus *short-term orientation*, which deals with how the culture views the past and the future. In a long-term focused culture, team development and a deep relationship with a client is very important, while in a culture that emphasizes the short-term, delivering a high-quality product on time is all that really matters.