PRACTICE OBJECTIVE

This practice will identify and prioritize the software attributes. Attributes are the quality characteristics of the software being tested. They are independent of the functional aspects, and primarily relate to the architectural structural aspects of a system. Attribute testing is complex and often avoided because it requires innovative testing techniques. However, many of the dissatisfactions of software by end users relate to attributes rather than functions.

The primary testing concern is that the system will perform functionally correct, but the quality and productivity attributes may not be met. The primary reason these attributes are not met is they are not included in most software specifications. For example, one frequently overlooked quality attribute is the ease of introducing changes into the software. This is a structural component, which is often left to the discretion of developers. Because implementing functionality takes a higher priority than attribute implementation in most organizations, the attributes are not adequately addressed.

PRACTICE WORKBENCH

The Quality Attribute Test Planning Workbench is illustrated in Figure 1. This figure shows that the input to the workbench is those quality characteristics which your organization believes can positively or negatively impact the quality of the operational software. Using that list of quality attributes, a determination is made by the involved parties (i.e., user, developers, testers) which quality attributes need to be incorporated into the test plan. The ones which will be tested need to be prioritized and the methods for validating those attributes, determined. The process is then checked to ensure the completeness of the workbench-produced quality attribute test plan.

KEY CONCEPT: IDEALLY, THE QUALITY ATTRIBUTES WOULD BE IDENTIFIED AND PRIORITIZED DURING REQUIREMENTS GATHERING. THE TESTER MAY OR MAY NOT PARTICIPATE IN THIS PROCESS.

INPUT PRODUCTS

The input into this testing practice is the list of quality attributes whose presence or absence can impact the quality of the operational software. A list of the more common quality attributes follows.

Correctness

Assurance that the data entered, processed, and outputted by the application system is accurate and complete. Accuracy and completeness are achieved through controls over transactions and data elements. The control should commence when a transaction is originated and conclude when the transaction data has been used for its intended purpose.
File Integrity

Assurance that the data entered into the application system will be returned unaltered. The file integrity procedures ensure that the right file is used and that the data on the file and the sequence in which the data is stored and retrieved is correct.

Authorization

Assurance that data is processed in accordance with the intents of management. In an application system, there is both general and specific authorization for the processing of transactions. General authorization governs the authority to conduct different types of business, while specific authorization provides the authority to perform a specific act.

Audit Trail

The capability to substantiate the processing that has occurred. The processing of data can be supported through the retention of sufficient evidential matter to substantiate the accuracy, completeness, timeliness, and authorization of data. The process of saving the supporting evidential matter is frequently called an audit trail.

Continuity of Processing

The ability to sustain processing in the event problems occur. Continuity of processing assures that the necessary procedures and backup information are available to recoup and recover operations should the integrity of operations be lost due to problems. Continuity of processing includes the timeliness of recovery operations and the ability to maintain processing periods when the computer is inoperable.

Service Levels

Assurance that the desired results will be available within a time frame acceptable to the user. To achieve the desired service level, it is necessary to match user requirements with available resources. Resources include input/output capabilities, communication facilities, processing, and systems software capabilities.

Access Control

Assurance that the application system resources will be protected against accidental and intentional modification, destruction, misuse, and disclosure. The security procedure is the totality of the steps taken to ensure the integrity of application data and programs from unintentional and unauthorized acts.

Compliance

Assurance that the system is designed in accordance with organizational methodology, policies, procedures, and standards. These requirements need to be identified, implemented, and maintained in conjunction with other application requirements.

Reliability

Assurance that the application will perform its intended function with the required precision over an extended period of time. The correctness of processing deals with the ability of the system to process valid transactions correctly, while reliability relates to the system being able to perform correctly over an extended period of time when the system is placed into production.

Ease of Use

The extent of effort required to learn, operate, prepare input for, and interpret output from the system. This testing factor deals with the usability of the system to the people interfacing with the application system.

Maintainable

The effort required to locate and fix an error in an operational system. Error is used in the broad context to mean both a defect in the system as well as a misinterpretation of user requirements.

Portable

The effort required to transfer a program from one hardware configuration and/or software system environment to another. The effort includes data conversion, program changes, operating system, and documentation changes.

Coupling

The effort required to interface one application system with all the application systems in the processing environment which either it receives data from or transmits data to.
Performance

The amount of computing resources and code required by a system to perform its stated functions. Performance includes both the manual and automated segments involved in fulfilling system functions.

Ease of Operation

The amount of effort required to integrate the system into the operating environment and then to operate the application system. The procedures can be both manual and automated.

IMPLEMENTATION PROCEDURES

There are three steps to execute to develop a test plan for the quality attributes of the software under test. The first is to define the attributes; the second is to prioritize the attributes. The third step is to determine how to test the quality attribute. The first is necessary because defining the attribute determines the levels of quality that the I/S function intends to implement in developed software. For example, if maintainability is one of the quality attributes, then the I/S standards need to state the structural characteristics that will be incorporated into software to achieve maintainability. The second is important because emphasizing one of the quality factors may in fact de-emphasize another. For example, it is difficult to have an operation that is very easy to use and yet highly secure. Ease of use makes access easy, while highly secure makes access difficult. The third step is difficult because there is a subjective aspect to testing quality attributes; for example, how do you test ease of use?

Step 1: Identify Attributes

The input document provides a representative list of quality attributes. Attributes must be adopted by the I/S function in order to be effectively tested. However, testers can use these attributes, and test against those attributes, even though the I/S function has not put into place the standards necessary to accomplish these attributes. The quality attributes to be tested should be listed on the Quality Attributes Test Planning Worksheet.

Step 2: Prioritize the Attributes

The attributes should be prioritized into three categories, as follows:

- Highly important to the success of the project
- Medium importance to the success of the project
- Low importance to the success of the project

The ranking may need to be done by the end user of the software. This is normally more desirable than to have testers rank the attributes. Post the attribute ranking to the Attribute Test Planning Worksheet. In the ranking column you can indicate an H, M, or L, for high, medium, or low importance. Also describe the impact the lack of this attribute could have on the software under test.

Step 3: Develop Test Conditions to Test Highly Important Attributes

Normally, the testers will not have time to test all of the attributes. Therefore, they should concentrate on the attributes that are of high importance to the software being tested. Also, the proper implementation of the attributes of high importance normally assures that the attributes of lower importance will also be adequately implemented. What is necessary to accomplish a few, contributes to the accomplishments of the others, and since the others are of lower importance their implementation does not have to be as well performed.

The test conditions to test the attributes should be listed on the Attribute Test Planning Worksheet.

CHECK PROCEDURES

Checklist contains items to help you determine that the list of test conditions you have created for the attributes testing process is complete. No responses indicate that you have not included a test condition of this type. You may want to indicate in the comments column the reason for not including this test condition. Note that there may be many transactions required to accomplish this specific test condition.
<table>
<thead>
<tr>
<th>ITEM</th>
<th>RESPONSE</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Have the software attributes been identified?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Have the software attributes been ranked?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Does the end user agree with the attribute ranking?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Have test conditions been developed for at least the high-importance attributes?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. For the correctness attributes, are the functions validated as accurate and complete?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6. For the authorization attribute, have the authorization procedures for each transaction been validated?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7. For the file integrity attribute, has the integrity of each file/subschema been validated?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8. For the audit trail attribute, have test conditions validated that each business transaction can be reconstructed?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9. For the continuity of processing attribute, has it been validated that the system can be recovered within a reasonable time span, and that transactions can be captured and/or processed during the recovery period?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10. For the service attribute, has it been validated that turnaround time/response time meets user needs?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11. For the access control attribute, has it been validated that only authorized individuals can gain access to the system?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12. For the compliance attribute, has it been validated that I/S standards are complied with, that the system development methodology is followed, and that the appropriate policies, procedures, and regulations are complied with?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13. For the reliability attribute, has it been validated that incorrect, incomplete, or obsolete data will be processed properly?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14. For the ease of use attribute, has it been validated that people can use the system effectively, efficiently, and economically?</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
CHECKLIST 1. ATTRIBUTES CHECKLIST (continued)

<table>
<thead>
<tr>
<th>ITEM</th>
<th>RESPONSE</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>15. For the maintainable attribute, has it been validated that the system can be changed/enhanced with reasonable effort, and on a timely basis?</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>16. For the portable attribute, has it been validated that the software can be moved efficiently to other platforms?</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>17. For the coupling attribute, has it been validated that this software system can properly integrate with other systems?</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>18. For the performance attribute, has it been validated that the processing performance/software performance is acceptable to the end user?</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>19. For the ease of use attribute, has it been validated that the operation personnel can effectively, economically, and efficiently operate the software?</td>
<td>YES</td>
<td>NO</td>
</tr>
</tbody>
</table>

DELEVERABLES

The deliverable from this workbench is the quality attribute test plan. A worksheet is provided to document the quality attributes component of the overall software test plan. This worksheet should be incorporated into the test plan document.

USAGE TIPS

The quality attributes test plan can be used in either of the following two ways:

* Educational - The quality attributes can be used to educate the development team as to the need to address these quality attributes; and to educate the users so that they will recognize that the developers and testers have taken into account the identified quality attributes.

* Creating test transactions - This component of the test plan can be used as input to the testers that will be developing the test transactions/test data to validate that the developed software meets the quality attribute requirements.
## QUALITY ATTRIBUTE TEST PLANNING WORKSHEET

<table>
<thead>
<tr>
<th>ATTRIBUTE</th>
<th>RANKING</th>
<th>IMPACT ON SOFTWARE</th>
<th>TEST METHOD</th>
</tr>
</thead>
</table>

Copyright © 1997 · Quality Assurance Institute® · Orlando, Florida