

Component 8 Installation and Maintenance of Health IT Systems

Unit 10 Developing a Test Strategy and Test Plan

This material was developed by Duke University, funded by the Department of Health and Human Services,
Office of the National Coordinator for Health Information Technology under Award Number IU24OC000024.

What We'll Cover

- What is User Acceptance Testing (UAT)?
- When to test?
- Testing strategy, scenarios, scripts
- What are the testing stages?
- Typical testing strategy template
- Test plans: utilizing resources
- Test scenarios vs. test scripts
- Creating test scenarios
- Test scripts
- Random testing

What is User Acceptance Testing (UAT)?

- Usually performed as final step before rollout
- Ensures product meets client's expectations
- Ensures software meets project objectives
- Provides opportunity for further debugging
- Based on mutually agreed-upon deliverables

When to Test?

- Prior to rollout
- Always needed for critical software such as EHR systems
- Focused on functionality, not technical issues. Most technical and integration bugs should have been worked out prior to testing.

What are the Testing Steps?

1. Planning the UAT
2. Designing test cases
3. Creating the UAT team
4. Executing test cases
5. Documenting defects
6. Resolving and debugging
7. Signing off

What are the Testing Steps?

- Three new terms

| Term | |
|----------------|--|
| Test Strategy | People, tools, procedures, and support |
| Test Scenarios | Situations to be tested |
| Test Scripts | Actual, step-by-step inputs that will be used, along with expected results |

Test Scenarios vs. Test Scripts

- Test scenarios
 - Broad interpretation of what you are testing
 - Example: “When you enter a prescription that could cause an adverse reaction for the patient, a warning message should appear.”
- Test Scripts
 - Step-by-step details of running the tests
 - Example: “1. Select John Doe’s patient record. 2. Select ‘New Rx’. ...”

UAT Planning Stage

- Most important stage in the process
- Testing strategy developed
- Key focus areas defined
 - Based on expected overall product deliverables
- Entry and exit criteria defined

Testing Strategy Template

- A typical testing strategy addresses several categories:
 - Overview
 - Testing environment
 - Procedures
 - Software
- The following slides show typical elements in a testing strategy.
- Your strategy may include more or fewer variables.

Testing Strategy Template: Procedures (cont'd)

| Category | Instructions |
|-------------------------|--|
| Sign-off for activities | How each activity in the testing will be signed off; includes both initial testing, outlined in test plan, and re-testing of defects that have been rectified. |
| Sign-off for project | How total testing will be signed off, including defect rectification process. |

Testing Strategy Template: Software

| Category | Instructions |
|---|---|
| Test management and performance management software | List any specialized test management software and manner of use |
| Testing software | Outline any software that will be used during the testing process |

Designing Test Cases

- Used to test focus areas defined during planning process
- Often defined during software requirements phase
- Others created by business analysts or subject matter experts

Testing Scenarios

- Broadly define scope of each test, with expected output
- Used to develop detailed scripts needed for end user testing
- Best generated by experienced testers and subject matter experts

Creating Test Scenarios: Example

Title: Data Input and Modification Validation Test
This test will validate the Secure User Access field.

| Test No. | Input Field | Input Type | Input | Anticipated Result | Notes |
|----------|-------------|--------------|--------------------|---------------------|--------------------------|
| I101 | User Name | Alphanumeric | Incorrect username | Error message | |
| I102 | | | Correct username | Prompt for Password | |
| I102 | | | None | Error message | Password cannot be blank |

Test Scripts

- Include:
 - Step-by-step instructions for end-user tester
 - Sections for recording actual output from tests
 - Instructions for passing along findings to appropriate team members for resolution
- Last stage before end user tester selection and testing begin

End User Testing Team

- Actual testing team should represent a cross-section of the end user environment.
- Testing environment should closely mimic the real production environment.

Test Scripts

- Test scripts can vary but usually include:
 - A column indicating the set number in the process
 - A column in the test script for the step by step instructions to perform
 - A column indicating the expected test result
 - A column for the tester to input the ACTUAL result
 - A comments field
- Sometimes, test scripts can be coded and automated using a tool called an interpreter to mimic the user.

Executing the Tests

- End users execute each of the test scripts, carefully documenting their findings.
- Also allow testers “free range” on the system to perform relevant random testing to explore for errors.
- Ensure all documentation is complete and forwarded quickly to proper team members for issue resolution.

Issue Resolution

- Issues discovered are discussed with the testing and project team &/or vendor.
- Development team &/or vendor devises satisfactory solution
- End user team retests.

Component 8/Unit 10 Health IT Workforce Curriculum
Version 2.0/Spring 2011 22

Sign-Off

- Acknowledgement that UAT team accepts the application
- All known issues have been satisfactorily resolved.
- Often represents pay-off point for vendor

Component 8/Unit 10 Health IT Workforce Curriculum
Version 2.0/Spring 2011 23

Summary

- Using talented resources to develop relevant test scenarios is critical to proper system validation and successful testing.
- Once test scenarios have been conceived, devise test scripts for each scenario.
- End users or automated programs perform the detailed step-by step testing and record the results.

Component 8/Unit 10 Health IT Workforce Curriculum
Version 2.0/Spring 2011 24

Summary

- Careful documentation and detailed defect resolution plans ensure kinks are worked out and retested.
- Issues are brought to the test team's attention to be forwarded to the programmers or vendor for resolution
- Testing continues until all parties are confident in the software's performance.
- Once testing is complete, the client "takes delivery" and can begin implementing its "roll out" strategy

Reference

- "Acceptance testing." Wikipedia.
– http://en.wikipedia.org/wiki/Acceptance_testing
- Neville Turbit. "Developing a Test Strategy." Project Perfect.
– http://www.projectperfect.com.au/info_test_strategy.php
- "Beginner's Guide to Software Testing."
– <http://www.freewebs.com/qualityqa/Complete%20Software%20testing.html>
