Oracle® AutoVue

Testing Guide

Release 21.1.0

F86915-01

September 2023

ORACLE

Testing Guide, Release 21.1.0

F86915-01

Primary Author:

Contributing Author:

Contributor:

Contents

Preface

The *Testing Guide* provides guidelines that should be followed when planning and performing Systems Integration Tests (SIT) and User Acceptance Tests (UAT) for an AutoVue and AutoVue for Agile deployment.

For the most up-to-date version of this document, go to the AutoVue Documentation Web site on the Oracle Technology Network (OTN) at https://www.oracle.com/technetwork/documentation/autovue-091442.html.

Audience

This document is intended for Oracle customers and partners who are deploying AutoVue and AutoVue for Agile in an enterprise environment. This guide serves as a good starting point for building your SIT and UAT scenarios to cover AutoVue functionality.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Documents

For more information about Oracle AutoVue, refer to the following documents on OTN:

- *Oracle AutoVue Planning Guide*
- *Oracle AutoVue Installation and Configuration Guide*
- *Oracle AutoVue Security Guide*
- *Oracle AutoVue Viewing Configuration Guide*
- *Oracle AutoVue User's Manual*
- *Oracle AutoVue Release Notes*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1 Introduction

This document covers what you need to know when performing Systems Integration Testing and User Acceptance Testing. This section will explain both of these deployment phases, followed by an overview of the remainder of the document.

1.1 Systems Integration Testing (SIT)

Systems Integration Testing (SIT) is performed by the systems integrator once AutoVue has been installed and configured. The goal of SIT is to verify that all the technical aspects of a software solution have been installed and configured properly and are working as expected. SIT will typically happen several times in a typical deployment, as there are multiple environments which need to be installed, configured, and tested.

1.2 User Acceptance Testing (UAT)

User Acceptance Testing (UAT) is performed by a select group of users on the staging environment, once SIT has been completed. The users are selected in such a way that they cover all the business needs to be addressed by the AutoVue solution. UAT allows the users to try AutoVue in a realistic environment, and is required for users to sign off that the solution meets their business needs.

1.3 Putting SIT and UAT in Context: Phases of a Deployment

In a typical AutoVue deployment there are five phases.

Phase 1: Solution Requirements

In the Solution Requirements phase, the business needs are identified.

Phase 2: Solution Design & Architecture

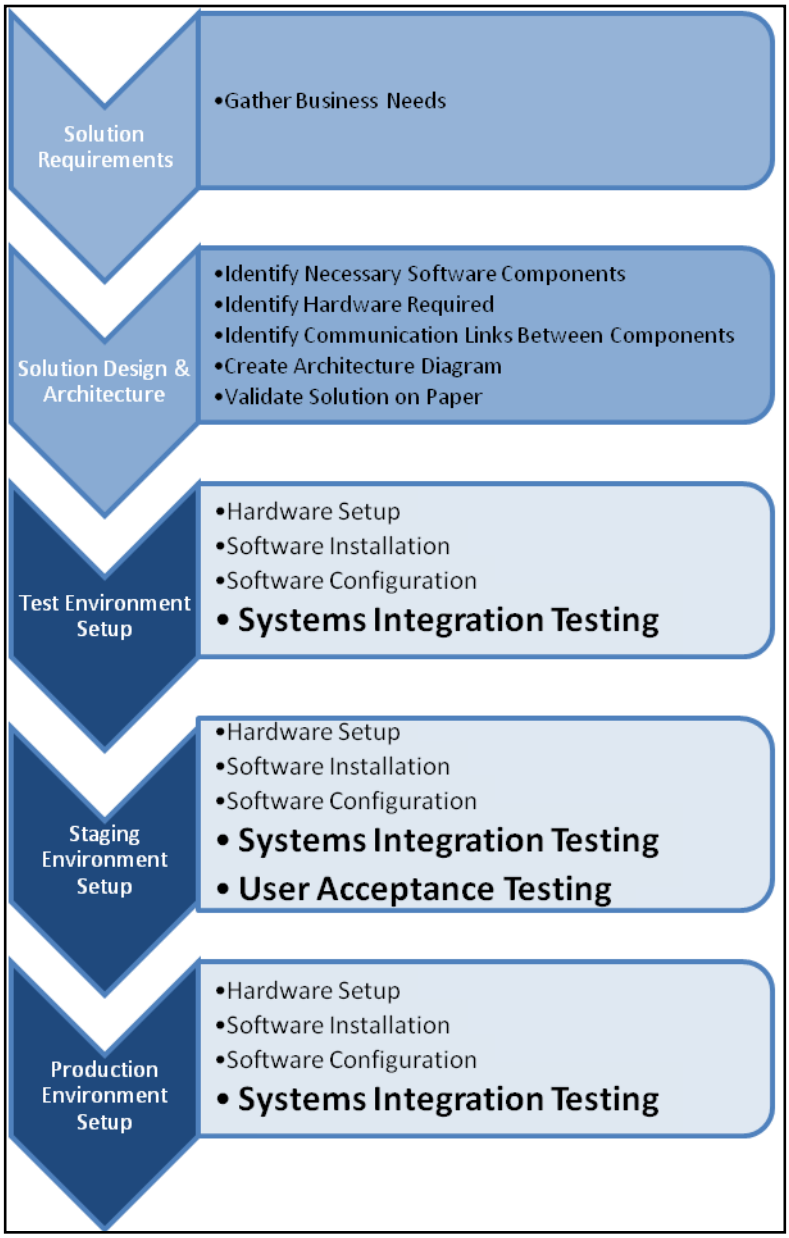
In the Solution Design & Architecture phase, a solution to the business problem is designed. All the necessary software components are identified, and the solution designer figures out how all these components will connect with each other. Hardware requirements such as the number of AutoVue servers and the number of application servers are identified. The entire architecture of the solution is documented in an architecture diagram showing all the hardware nodes, where each software component sits, and how the components communicate with each other. The solution is validated on paper to ensure all the software components are compatible with each other, with the operating systems, and with any other factors affecting the architecture (such as firewalls, geographic locations and load balancers).

Phase 3: Test Environment Setup

In the third phase a test environment is set up to test the solution. The test environment is recommended to be as close as possible as the final production environment, but there may be

some differences such as fewer server nodes, due to cost. Machines are set up for each node in the test environment, and the appropriate software is installed and configured on each node. At this point Systems Integration Testing is done by the systems integrator to ensure that AutoVue works properly in this environment. Any technical issues should be worked out in Phase 3 before moving on.

Figure 1–1 Phases of a Development



Phase 4: Staging Environment Setup

Phase 4 is similar to the test environment setup. Since this will be the final testing environment before going live, the staging environment is set up to be as close as possible to the final production environment. As in Phase 3, once the hardware and software have been set up, Systems Integration Testing is done by the systems integrator to ensure all the technical components are functioning as they should. Finally User Acceptance Testing (UAT) is done by

a set of users, who will try out the solution to ensure it meets their business needs. After the users sign off on the UAT, Phase 5 can begin.

Phase 5: Production Environment Setup

Setting up the production environment is the final stage of the deployment. All technical issues with the solution have been worked out, and the users have signed off on the solution meeting their business needs. The actual production hardware nodes are setup, and the software is installed and configured on each node. A final round of SIT is done to ensure no mistakes were made during installation and configuration. Once this round of SIT is done, the system goes live with users.

1.4 Differences Between UAT and SIT

A common point of confusion is mixing up User Acceptance Testing (UAT) and Systems Integration Testing (SIT). Systems Integration Testing is to be performed by a systems integrator, and its goal is to validate that all technical aspects of the solution are working. By contrast, UAT is performed by a set of actual users of the system. The goal of UAT is to validate that the solution meets the users' needs and solves the original business problem as intended.

Several, though not all, of the tests that will be performed will be common to both SIT and UAT; however, it is important to conduct both SIT and UAT. Systems integrators can't answer whether the solution meets business needs, and users should not be testing a system that has not first been validated by a systems integrator to work technically.

Table 1–1 Differences Between UAT and SIT

	User Acceptance Testing	Systems Integration Testing
Deployment Phases	Phase 4: Staging Env. Setup	Phase 3: Test Env. Setup Phase 4: Staging Env. Setup Phase 5: Production Env. Setup
Performed By	Select group of users	Systems Integrator
Goal	Ensure solution meets business needs	Ensure technical aspects of solution works
Preceded By	SIT	Software Installation and Configuration
Followed By	User Base Sign Off	UAT
Consists Of	User Scenarios	Individual Test

1.5 Overview of this Document

This document explains what should be considered when planning SIT and UAT for AutoVue and AutoVue for Agile. It briefly covers how to fit SIT and UAT into your deployment process. Then it goes through some of the features of AutoVue, and what types of tests need to be performed to cover these features. The document also gives a short sample test plan to use as a starting point when developing your own test plans. Finally, some sample UAT test scenarios are presented to use as a guide when developing your own scenarios.

After reading this guide you should be able to create a series of systems integration tests to perform on your environment. Performing these tests on each environment will ensure that AutoVue, and all its components have been installed and configured properly, and work with any other components AutoVue has been integrated with.

Similarly, after reading this guide you should be able to create a set of user acceptance testing scenarios for your environment. Having a select group of users run through these scenarios in a staging environment will allow them to sign off on the solution meeting their business needs.

1.6 Terminology

Throughout this document the term DMS will be used to refer to all back-end systems that can be integrated with AutoVue such as Document Management System (DMS), Product Lifecycle Management (PLM), Enterprise Content Management (ECM), Enterprise Resource Planning (ERP), and so on.

2

SIT Process Requirements

This section covers some factors that should be taken into account when deciding how SIT should be organized and scheduled.

2.1 How Many Rounds of SIT

For larger deployments, there will be three different environments that will need to be configured. Each environment will need to undergo SIT.

2.1.1 Test Environment

It is recommended to plan at least two rounds of SIT for the test environment. This way, if configuration problems or product defects are found in the first round, there will be a second round to validate that any deployment changes have fixed the problem and have not introduced any new problems.

2.1.2 Staging Environment

If the staging environment is identical in architecture to the test environment, then only one round of SIT may be necessary. However, it is common to have a test environment which lacks many of the features of the production and staging environments, such as load balancers, firewalls, proxy servers, and so on. If any of these or other significant changes are present between the staging and production environments, then it is recommended to schedule at least two rounds of SIT for the staging environment.

2.1.3 Production Environment

Since most of the technical problems will be worked out in the testing and staging environments, it should not be necessary to perform more than one round of SIT for production. Note, however, that this requires that the staging environment be almost identical in architecture to the production environment. It is not recommended to introduce significant architectural changes at this stage.

2.2 Schedule

It is important that SIT is scheduled properly in the deployment process. Enough time must be given for each SIT phase to be completed. Between each phase of SIT, there should be enough time to change the configuration and respond to product defects that were caught in the previous phase before starting the next one.

2.3 Test Plan

To ensure that all required tests are performed, it is recommended that a test plan be created. This often takes the form of a checklist, which can be completed as each test is performed. The test plan consists of individual tests as opposed to complex user scenarios. Each test should be designed to test a particular feature, connection, or element of the configuration. For information on how to design these tests, refer to Chapter 4, "AutoVue Testing Guidelines."

3

UAT Process Requirements

This section covers some factors that should be taken into account when deciding how UAT should be organized and scheduled.

3.1 How Many Rounds of UAT

It is recommended to plan at least two rounds of UAT for larger deployments. This way, if changes need to be made to address user concerns, there will be a second round to validate these changes.

3.2 Schedule

It is important that UAT is scheduled properly in the deployment process. Enough time must be given to each UAT phase to be completed. Between each phase of UAT, there should be enough time to change the configuration and respond to user concerns found in the previous round before starting the next one.

3.3 Scenarios

It is important that UAT consists of complete user scenarios as opposed to individual tests. As an example, the users' business process may be that files are opened, measured, and then a markup is added and stored back into the DMS. This could be tested by repeating that process from start to finish, or by performing isolated tests. The problem with performing isolated tests is that some of the process will not be covered unless the process is tested from start to finish. Example: Given the above business process, say the following tests are performed in isolation:

1. The file is opened
2. The file is measured
3. A markup is created and stored in the DMS

Let us say that all 3 of these tests were performed at different times. It is possible that something has been missed here. What if the markup created in step 3 was not a measurement markup (which is what we would expect given the business process)? Any issues with the solution when storing measurement markups may not be found. Creating test scenarios that follow the business process flow ensures that even small details that may not have been considered when planning the tests will be covered.

3.4 User Training

It is recommended that the users that will take part in UAT are familiar with AutoVue. Testers who are not familiar with AutoVue will need to be given test scripts with very particular

instructions on how to perform various tasks. The risk of giving very detailed instructions is that the tests may not actually test how the users would interact with the system naturally. Recommended reading is provided in "Related Documents".

4

AutoVue Testing Guidelines

The following guidelines should be used when creating SIT tests or UAT tests. When creating SIT tests, also be sure to take into consideration the SIT specific guidelines in the next section.

4.1 Internal / External Users

While testing, it is important to connect to AutoVue over the network in the same way that the users will. There are generally three possibilities:

- Users connect over the local intranet.
- Users connect from outside the network (usually through a firewall).
- Both of the above.

In the case where both internal and external users are going to be using AutoVue, it is important to perform the tests both from inside the network and from outside the network. Firewalls, security protocols (SSL), proxy servers, DMS or AutoVue customizations can all be factors which may cause either a scenario to pass internally and fail externally or vice versa.

4.2 Environment / Architecture

It is important to test AutoVue in an environment as close as possible to the production environment. Factors to consider are:

- **Version of Java Runtime Environment (JRE) used by the clients:** Different JREs can exhibit slightly different behavior. Ensure the same version of JRE that will be used in production is being used during testing. It is recommended to use one of the JRE versions certified by Oracle. For information on certified JRE versions, refer to the "System Requirements" section of the *Oracle AutoVue Client/Server Deployment Installation and Configuration Guide*.
- **Multiple AutoVue servers**: If production will have multiple AutoVue servers for failover or load balancing, the test and staging environments should also have multiple AutoVue servers configured in the same way.
- **Multiple DMS instances** (if applicable): If production will have multiple instances of a DMS, it is important that the test and staging environments also have this. For SIT, this is particularly important, as it will ensure the load balancing of the VueLinks and DMS is working correctly, and will catch any problems with the configuration between the DMS, the VueLinks, and AutoVue.
- **Installed Software:** It is important to make sure any software that will be installed in production on any of the DMS, Application Server, or AutoVue machines, is installed and configured in the same way in the test and staging environments.

- **Printer Configuration:** Ensure printers connected to the test/staging AutoVue server machines are configured the same way as the printers that will be connected to the production AutoVue server machines.
- **Fonts:** Ensure the same fonts are installed on the test/staging AutoVue server that will be used in production.
- **Permissions:** Ensure that the same user permissions are configured on test/staging environment that will be used in production.
- **Mapped drives:** In the event mapped drives are used, you must ensure that the same drives are accessed in test/staging environment that will be accessed in production.

4.3 Sample Designs

A large part of testing AutoVue involves viewing sample designs. Here are some guidelines to use when selecting these designs.

4.3.1 Format Selection

AutoVue supports many different formats and each format type has functionality and behavior which is specific to that type. Similarly, the AutoVue functionality used to view one particular format is often only used to view that specific format. To ensure all the appropriate functionality is used during testing, make sure each format that will be used is part of the test scenarios. If you are upgrading any of your tools used to create documents/designs, make sure documents created with these new tools are also tested as part of the AutoVue SIT and UAT phases. For more information on supported formats, refer to the *Supported File Formats* document on the Oracle Technology Network (OTN) https://www.oracle.com/technetwork/documentation/autovue-091442.html.

4.3.2 Realistic Designs

When testing, it is important to use designs that will actually be part of everyday AutoVue use. Using overly simple designs will not result in a realistic test of the system. For example, when selecting a test design, you should take into consideration the file size, how the file was generated (important when opening formats such as PDF and DWG) as well as the format type.

4.3.3 Format Fidelity

One of the major factors that needs to be tested during AutoVue testing is format fidelity. For each design format used in your business process, test enough designs to ensure there are no issues displaying these files.

4.4 Integrated Tests

When AutoVue is integrated with a DMS or PLM system (e.g., Oracle Agile, Oracle UCM, Documentum, and so on.), there are specific use cases that need to be tested. These are some of the things that should be considered when testing AutoVue in such an environment.

4.4.1 Open Designs

When AutoVue is integrated, it is important to test opening designs through the DMS's customized GUI. Two different scenarios should be tested:

- Open a design found by browsing the repository
- Perform a search in the repository and open a design from the search results

4.4.2 Comparing Versions

A common use case to test is to open two different versions of the same design and compare them in AutoVue.

4.4.3 Creation of Stamp Markup

Stamp is a markup that is only available in integrated environments. A Stamp entity displays attributes that come from the DMS itself such as the last time the file was saved, who approved it, and so on.

SIT: Testing Stamps is important as it will indicate whether the integration between AutoVue and the DMS is set up correctly, and will also determine whether the fields required by the Stamp entity match those found in the DMS.

UAT: It is important to test Stamps to ensure that the stamp design and DMS fields that are exposed meet the needs of the users.

4.4.4 Renditions

If the VueLink or DMS integration is configured to generate renditions (such as thumbnails), the following tests should be performed:

Ensure renditions are being generated and look like the original design.

SIT: The following rendition tests should be performed in an SIT phase only:

- Ensure renditions are being created in the right location.
- Ensure renditions are being created with the right permissions.

4.4.5 Customizations

Any customizations done to the DMS involving AutoVue should be tested. Examples of such customizations include:

- Restricting access to viewing with AutoVue based on user.
- Buttons added to the DMS to initiate viewing
- Any DMS scripting that triggers AutoVue (for example, to trigger converting a file to PDF)

4.5 Markups

The following sections discuss guidelines for testing markups.

4.5.1 Creation and Saving of Markups

Make sure users can create and save markups.

4.5.2 Opening Saved Markups

Make sure users can open the markups they have created. When opening a design that has markups, make sure the markup icon appears in the lower left corner of the AutoVue window.

4.5.3 Markup Permissions

When integrated into a DMS system, permissions can be set to determine who can create and open markups. Tests should be created to ensure these permissions are working correctly. Example:

For each user in a particular role:

- They can access each other's markups
- They can read the markups but not modify them
- They can create a master markup

4.5.4 DMS Markup Storage

When you are integrated into a DMS, the following markup tests should be performed during SIT:

- Ensure markups are being stored in the correct location
- Ensure markups are being stored with the correct permissions

4.6 Performance

It is important to test AutoVue performance to ensure it is acceptable to the users. There are a few tests that should be considered:

- **Everyday Viewing Performance:** Ensure that AutoVue performs acceptably when viewing designs of average size.
- **Large Design Performance**: Ensure AutoVue performs acceptably when viewing the largest possible designs the user base is expected to encounter. It is worth noting that since these designs may not be encountered frequently, what is considered acceptable for these designs may be different from the criteria to evaluate performance for everyday files. (Perhaps 5 seconds is acceptable for everyday designs, and 30 seconds is acceptable for huge designs)
- **Load Testing (SIT Only)**: If possible, it is a good idea to test an AutoVue server with the same amount of load that would be expected in production. This type of test is not always possible since doing so requires many clients accessing the server at one time. Performing these tests can detect any performance problems due to overloaded servers. A valid load test case involves having multiple clients on different machines connecting to the server at the same time (open sessions). Additionally, the number of active sessions, where there is communication between the client and server, affects load testing. Note that having multiple clients on the same machine connecting to the AutoVue server is not a valid test case. When possible, load testing should involve both internal clients and external clients.

4.7 General Functionality

The following sections list guidelines to test the general functionality of AutoVue.

4.7.1 Printing

Print a design to the printer from the AutoVue interface. Be sure to test the following:

- Print designs with markups.
- Print using paper sizes that will be typical of the users' needs (e.g., whether they are printing to letter or E1).

- Test headers and footers (if applicable).
- Test pen settings (if applicable).

4.7.2 Measurements

Users who will be using raster images, 2D, 3D, or EDA designs usually perform measurements as part of their daily process. If this is the business process in your environment, testing measurements should be part of SIT and UAT. Make sure measurement values and units match what is expected.

4.7.3 Design Conversion

The following should be considered when designing tests to convert designs to other formats. Tests should check that:

- The design was converted correctly (when viewing, it looks similar to the original design).
- The converted file gets checked into the DMS and is accessible (if applicable).

4.7.4 Customized GUI

Some deployments of AutoVue include customizations to the user interface (through a custom GUI file) to modify which features of AutoVue are accessible and modify the layout of those features. Test scenarios should include tests to check that:

- Users are seeing the appropriate customizations based on the user's role (some user groups may use different GUI files).
- Any features that should be hidden are not accessible.
- Any custom features are accessible.

It is important to note that user interface can be modified for each of the different viewing modes. Therefore it is important to check the above in different viewing modes where a given feature might apply. Example: If printing functionality should be hidden, this should be tested when a 2D file is opened, a 3D file is opened, a markup file is opened, and so on. For more information on different viewing modes, refer to the "Role-Based GUI" chapter of the *Oracle AutoVue Client/Server Deployment Installation and Configuration Guide* on OTN.

4.8 Real-Time Collaboration Functionality

Any collaboration features that will be part of the business process should be part of the SIT and UAT. If Real-Time Collaboration sessions are going to be used in production, design some tests to make sure a Real-Time Collaboration session can be established.

4.9 Format Specific Functionality

Different format types have functionality in AutoVue which is specific to that particular format type. Below is a list of tests that should be done based on the types of files that will be viewed.

4.9.1 2D-Specific Functionality

The following is a list of features that should be part of 2D test scenarios.

- Ensure all fonts are picked up correctly.
- Ensure all XRefs are loaded correctly.
- Print with pen settings.

- Modify layer settings.

4.9.2 3D-Specific Functionality

The following is a list of features that should be part of 3D test scenarios.

- Ensure all parts are loaded correctly.
- Select parts and display their properties.

4.9.3 EDA-Specific Functionality

The following is a list of features that should be part of EDA test scenarios.

- Select entities and display their properties.
- Cross probe between schematic and PCB.
- Turn on and off PCB layers.

5

SIT-Specific Guidelines

The following guidelines should be considered when creating SIT tests, but should not be included in UAT. These tests should be done in addition to the tests in the previous section.

5.1 Upgrades

If this deployment is an upgrade to an existing AutoVue environment, the following tests should be performed:

5.1.1 Open and Resave Old Markups

Using the new version of AutoVue, open up markups created with the old version of AutoVue. Text markups are particularly important to test. Make sure all the information is visible. Edit the markup and resave it. If the markup is closed and reopened, ensure that the markup information is still there.

5.1.2 Cached Files

Tests should be done to make sure that files that have already been opened with the previous version of AutoVue and have been cached on the AutoVue server open correctly with the new version of AutoVue. Note that these files will not be loaded from streaming files.

5.1.3 Configuration

Tests should be done to ensure that any configuration settings that were set in the previous setup are still working as expected. This includes changes that have been made to:

- Jvueserver.properties
- Allusers.ini
- default.ini
- ini settings stored in the Profiles directory
- custom GUI files
- Markup symbols
- Logging configurations

Note: The format of some of these files and options changes from version to version of the software. Simply ensuring that the old configuration files are in place in the new deployment is not enough to ensure the users will get the expected behavior. For more information on configuration settings, refer to the *Oracle AutoVue Client/Server Deployment Installation and Configuration Guide* on OTN.

5.2 Streaming Files

AutoVue is typically configured to use its streaming file technology to improve the performance of viewing designs and documents. SIT should include tests for the following:

- Verify whether when opening a design for the second time that it is being loaded from the streaming file.

If Integrated with a DMS

- Verify whether the streaming file is being stored in the correct location. For example, if it is configured to be stored in the DMS.
- Verify that the permissions on the markups within the DMS are correct. For example, whether users other than the markup owner can modify it.

5.3 Patches

If any patches have been installed to address bugs or to provide custom functionality, it is important to develop tests to verify the features provided by the patches. These tests can help determine whether there is an issue with the patch and help detect problems (such as a patch not being installed).

6

AutoVue for Agile Specific Testing Guidelines

This chapter discusses the testing guidelines for features that are specific to AutoVue for Agile solution. It covers both UAT and SIT tests.

Note: The testing scenarios that are covered in the previous chapters are applicable to the AutoVue for Agile solution as well.

6.1 Testing Guidelines

The following guidelines should be used when creating SIT tests or UAT tests.

6.1.1 Agile PLM Markup Storage

In an Agile PLM system, the following markup tests should be performed during SIT:

- Ensure markups are being stored in Agile vault.
- Ensure markups are being stored with the right permissions

6.2 General Functionality

The following section lists the guidelines to test the general functionality of AutoVue for Agile.

6.2.1 View Files in Vault

View the local file in the AutoVue viewer window and try to zoom, fit, pan and rotate the different formats like 2D and 3D files.

For 2D files, verify that the drawing is rendered correctly from the selected view-point. Ensure that you can compare the AutoCAD files without the viewports.

For EDA files, view the net connectivity of an entity such as a pin, via, or trace.

In case of 3D files, verify that the default, standard, and native views are displayed correctly during the testing phase. Also, define views using the User-Defined View feature and see whether the views are displayed correctly. Design test cases to view a model from a particular point using the Camera Views feature. Use the Re-center option to reposition the model back to the center of the viewing window.

Design test cases to check out, check in or view a single or multiple attachment files that are stored in the Agile File Vault, part of Agile File Manager (AFM). Each attachment file in the vault is associated with a numbered Agile file folder object. Re-use the attachment files, and design test cases that check whether there is proliferation of multiple copies of the same file in the file vault.

Also, use the Agile Browse and Agile Search menu items to browse and search for the required files.

Note: Agile Objects have to be marked in Agile A9 for Agile Browse functionality.

6.2.2 Comparing Documents

Verify that when you compare two files, the color-coded comparative data is displayed correctly. Three windows have to be displayed: right window has to display the newer version of the document, left window has to display the older version of the document, and the bottom window has to display the comparison results. Test the compare functionality by browsing and searching for files through the Agile Browse and Agile Search features.

Note: The Compare feature performs only a graphical comparison not a geometrical comparison.

6.2.3 Overlay

Verify whether you can overlay other files with the current active file. Test the application for overlay functionality by adjusting, and moving the overlay. You can even try to scale the overlay for defined X and Y co-ordinates.

Test the overlay functionality by browsing and searching for files through the Agile Browse and Agile Search features.

6.2.4 Cross Probe

Cross Probe between two or more EDA files. The selected entities in the schematic drawing or the PCB design is highlighted in all the open files. Test that the Cross Probe feature lets you select entities to highlight in the 2D and 3D views of the same file.

Test the cross probe functionality by browsing and searching for files through the Agile Browse and Agile Search features.

6.2.5 Redlining Attachments

Design some tests that add redline layers to attachment files. Add your comments, notes, and drawings as part of redlining to the attached files. Save the redline attachments while cross probing and comparing operations of the files. All markups are saved in a separate file called a Markup or a Markup file. When you display a file with its markups, the markups are overlaid as a layer on top of the drawing. AutoVue provides circles, clouds, polygons, and leaders as a variety of flexible, user-friendly entities. Attach text to entities, insert a note for longer comments, add attachments, or add a stamp such as a company logo. Create measurement markup entities and hyperlinks that link between the current file and other associated files or applications.

6.2.6 Augmented Business Visualization Functionality

Test whether you can create change objects such as Engineering Change Request (ECR) and Engineering Change Order (ECO), directly from the viewing document in AutoVue. Augmented Business Visualization (ABV) is a visualization framework that connects portions of documents to business data found in enterprise applications. Use ABV's hotspot capabilities to create links between objects in AutoVue's data model and objects in an external system. Test this feature by clicking an area of a document in AutoVue, which triggers a visual action and/or

the display of information is displayed in other applications. Verify that with visual dashboards, you can expose data from enterprise systems visually by changing the hotspot color.

6.3 SIT-Specific Guidelines

The following guidelines should be considered when creating SIT tests, but should not be included in UAT. These tests should be done in addition to the tests described in the General Functionality section.

6.3.1 Testing System Requirements

Verify the system requirements of the AutoVue for Agile installation, and ensure that it meets the requirements as described in the *Oracle AutoVue for Agile Installation and Configuration Guide*.

6.3.2 Testing AutoVue Client Libraries

Ensure that you have selected "Upgrade Client Libraries" option if you are running the AutoVue for Agile PLM installer on an Application Server or File Manager. After the installation, you have to verify that the application.ear file and the webfs.war file were redeployed correctly.

6.3.3 Testing File Manager Configuration Page

Check the File Manager Configuration to see whether the File Manager Connection, Application Server Connection and the VueServer Connections are successful. The Table 6–1 displays the settings, the value and the status of the connection. Verify if the connections were successful.

Note: Refer to *Configuring File Manager* in the *Oracle Agile Product Lifecycle Management Application Installation Guide* for File Manager Configuration settings.

Table 6–1 File Manager Configuration Page

Setting	Value	Status
File Manager Connection	http://example.com/Filemgr/AttachmentServlet	Success
File Manager Connection	http://example.com/Filemgr/services/FileServer	Success
Application Server Connection	http://example.com/Agile/PLMServlet	Success
Application Server Connection	http://example.com/Agile/services/FSHelper	Success
VueServer Connection	http://example.com/Filemgr/VueServlet	Success
VueServer Connection	http://example.com/Filemgr/VueLink	Success
VueServer Connection	http://example.com/Agile/services/DmsService	Success

7

Sample SIT Test Plan

The following is only a small sample of a test plan to illustrate what one should look like. It is far from complete, and should not be used as is for a complete SIT test plan. Tests should be created to cover all the aspects of your particular environment, using the previous sections as a guide.

Table 7–1 Connecting/Launching

Test All Expected Results are Yes unless otherwise indicated	Result From Internal Client	Result from External Client
Can you connect to the VueServlet using the app server name and port? (http://appserver:8080/servlet/VueServlet) Expected: Yes for Internal, No for External.		
Can you connect to the VueServlet using the proxy URL? (http://www.mycompany.com/AutoVue/servlet/VueServlet)		
Can you launch AutoVue?		

Table 7–2 GUI File Customizations

Test All Expected Results are Yes unless otherwise indicated	Result From Internal Client	Result From External Client
Are users able to print 2D files? sample: C:\samples\testfile1.dwg Expected: No		
Are users able to print 3D files? sample: c:\samples\testfile2.prt Expected: No		

Table 7–2 (Cont.) GUI File Customizations

Test All Expected Results are Yes unless otherwise indicated	Result From Internal Client	Result From External Client
Are users able to print PDF files? sample: c:\samples\testfile3.pdf Expected: No		
When viewing 3D files does the model tree appear at the left or the right of the screen? sample: c:\samples\assemblies\testasm1 .asm Expected: Right		

Table 7–3 Integration with DMS

Test	Result From Internal Client	Result From External Client
View a file by browsing the DMS		
View a file from a DMS search result		
Save a markup		
Can users in the Reviewer group view markups? sample: c:\samples\testfile1.dwg user: testreviewer Expected: Yes		
Can users in the Viewer group view markups? sample: c:\samples\testfile1.dwg user: testviewer Expected: No		

8

Sample UAT Scenarios

The following are some sample scenarios to give an idea of what a typical user scenario should consist of. Each scenario involves several tests which are not described in detail, but should be fleshed out in your own UAT test set. Note that we begin with use case scenario of a typical series of actions a user is expected to perform. This scenario is then developed into a test plan involving a specific set of tests.

8.1 AVScenario1 - Internal User Review

User Andrew works for Company X. He is working in Agile, and accessing it within CompanyX's intranet. He browses Agile and finds the AutoCAD design ABC. Andrew opens the design in AutoVue through the Agile GUI. Once open, he compares ABC with the previous version of ABC saved in Agile. He notices that part of the design has changed that shouldn't have. One of the walls is too long. Going back to the latest version of ABC, he creates a measurement markup to show that the wall is too long and stores this back in the DMS.

Test Plan:

AVScenario1 Test Plan

Table 8–1 AVScenario1 Test Plan

Test **For all tests use the following:** **user: Andrew** **sample: design ABC, revision 51** **browser from within the firewall**	**Result From Internal Client**
Open design ABC from Agile by browsing for it.	
Compare current revision of ABC with previous revision. Expected: Differences shown.	
Create measurement markup by measuring wall. Expected: Markup gets created correctly. Expected: Measurement is correct length.	
Save markup Expected: User has permission to save it.	

8.1.1 AVScenario2 - External User Post Review Feedback

Scenario: User Bill works for ContractorY, who is contracted to do some work for CompanyX. Bill is accessing Agile through the internet. Andrew told Bill there was a problem with design

ABC. Bill searches Agile for design ABC. From the search results, Bill opens design ABC. He notices that there are markups attached to the design. Bill opens the attached markup that Andrew created. Bill tries to print the design and markup but notices that the Print functions have been removed from the GUI. Bill tries to modify Andrew's markups but does not have permission.

Tests Involved:

- Open from Agile Search Results
- Open markup created by someone else
- Ensure Printing is disabled for external users
- Users not allowed to change someone else's markups

Table 8–2 AVScenario2 Test Plan

Test	
For all tests use the following:	
user: Bill	
sample: design ABC, revision 51	
browser from outside the firewall	**Result From External Client**
Search for design ABC in Agile.	
Open AutoVue from the search results.	
Open the markup attached to the design	
Expected: Bill should have permission to open the markup	
Print the design	
Expected: Bill should not be able to print. Print should not appear in the user interface.	
Make changes to the markup and save.	
Expected: Bill should not have permission to do this.	

A

Feedback

If you have any questions or require support for AutoVue, please contact your system administrator. If the administrator is unable to resolve your issue, please contact us using the links below.

A.1 General AutoVue Information

Web Site	https://www.oracle.com/us/products/applications/autovue/index.html

A.2 Oracle Customer Support

Web Site	https://www.oracle.com/support/index.html

A.3 My Oracle Support AutoVue Community

Web Site	https://communities.oracle.com/portal/server.pt

A.4 Sales Inquiries

E-mail	https://www.oracle.com/corporate/contact/global.html