

Testing Your Workday Deployment

The definitive guide



Contents

- 1. Overview** **4**
 - 1.1 Background 4
 - 1.2 Who Is This Guide For? 4
 - 1.3 Introduction to Testing Workday and Test Strategy 5
 - 1.4 Roles and Responsibilities 5
- 2. Testing Methodology** **6**
 - 2.1 Workday Deployment 6
- 3. Testing Preparation** **7**
 - 3.1 Roles and Responsibilities 7
 - 3.2 Activities and Deliverables 7
- 4. Configuration Review** **9**
 - 4.1 Roles and Responsibilities 9
 - 4.2 Configuration Review Entry Criteria 10
 - 4.3 Activities 10
 - 4.4 Configuration Review Exit Criteria 12
- 5. The Test Stage** **13**
 - 5.1 Test Case Preparation 13
 - 5.2 Roles and Responsibilities 13
 - 5.3 The Test Stage Entry Criteria 14
 - 5.4 Functional Testing 15
 - 5.5 Integration Testing 15
 - 5.6 Security Testing Activities 18
 - 5.7 End-to-End Testing Activities 18
 - 5.8 The Test Stage Exit Criteria 19

6. User Acceptance Testing	20
6.1 Roles and Responsibilities	20
6.2 UAT Entry Criteria	21
6.3 UAT Testing Activities	21
6.4 UAT Exit Criteria	22
7. Payroll Parallel Testing	23
7.1 Roles and Responsibilities	23
7.2 Payroll Parallel Testing Entry Criteria	24
7.3 Payroll Parallel Testing Activities	24
7.4 Sample Payroll Parallel Test Plan—Steps Through Testing	25
7.5 Payroll Parallel Testing Exit Criteria	26
8. Tenant Strategy	27
9. Defect Management	28
9.1 Defect Lifecycle	28
9.2 Defect Severity Criteria	29
9.3 Defect Priority Criteria	30
10. Test Reporting	31
11. Automated Testing	32
12. Glossary of Terms	35

1 Overview

1.1 Background

This is a test strategy guide for Workday deployments. It's been compiled by our deployment and Kainos Smart test teams based on our involvement in the testing of over 200 Workday deployments.

In it we've detailed the elements that you need to address when developing a testing strategy for your Workday Deployment. If you have any comments or questions about this guide, or would like help with your Workday testing strategy, please contact us via our website www.kainosworksmart.com or by using the contact details on the back cover.

1.2 Who Is This Guide For?

We've written the guide for those with primary responsibility for your overall Deployment Project and its testing activities. This is often the Project Manager and Test Manager. However, anyone on your Project Team can benefit from the information it contains. Note that some of the terminology we use in the guide is Workday specific, and we've assumed that you have a high-level understanding of the stages of a Workday deployment.

It's important to note that every Workday deployment is different, and as a result we've only included sections that are common across all Workday deployments. It's the responsibility of your project's Test Manager to use the advice in this guide to create your test strategy and to add to it if you have additional areas or testing activities that we've not covered.

Also note that we haven't described Data Validation in this guide. This is an important supplemental activity to Testing that requires its own strategy and planning. Data Validation needs to take place for each new Workday tenant. That activity will be shaped by the data being loaded to Workday.

1.3 Introduction to Testing Workday and Test Strategy

Workday is a cloud-based product where every customer is on the same version of the software. During the early stage of your deployment project, you'll make decisions on how you want and need Workday to be configured to meet your business (and possibly legislative) requirements. This includes how Workday interacts with your third-party and internal systems.

The reason for testing is not to test the Workday product itself, but to test that your instance of Workday has been configured for your specific requirements. Testing throughout your deployment project will give you the confidence that your configuration is fit-for-purpose and ready for live operation.

Your Workday Testing Strategy outlines the testing approach your organisation will use during deployment, including the scope, timeline, roles and responsibilities, tools, processes and standards you'll use during each test stage. It acts as the reference point for all members of the Project Team and provides information on what will be tested, and how, during your deployment.

1.4 Roles and Responsibilities

We've outlined the roles that will be involved in each of the testing stages that you'll need to complete during your deployment. Depending on your organisation, you may already have full-time staff in some of these roles. If not, you'll need to rely on Subject Matter Experts (SMEs) from your HR, Finance and Payroll teams to take on these roles.

2 Testing Methodology

2.1 Workday Deployment

Workday's Deployment Methodology consists of five distinct phases: Plan, Architect, Configure and Prototype, Test, and Deploy. Test planning and preparation occurs during the Planning and Architect stages, with the key testing activities taking place during the Configuration and Prototype and Test stages of the methodology.

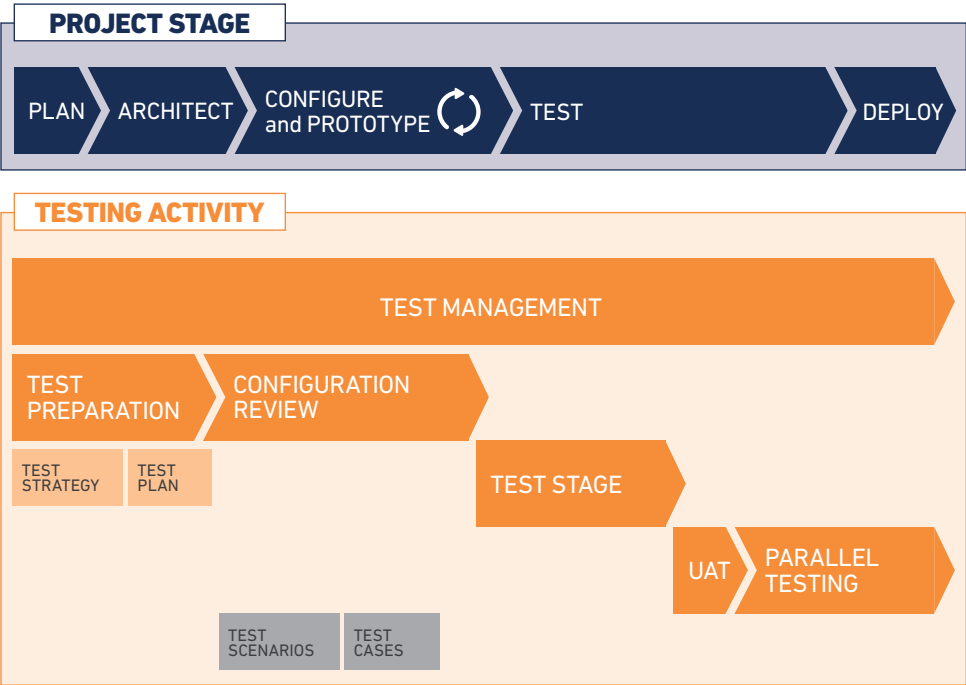


Figure 1—Workday Deployment Stages and Testing Activities

The key testing activities outlined in the illustration above are detailed in sections 3 through 7.

3 Testing Preparation

The purpose of this phase is to plan and organise the testing activities during your Workday deployment. The key artefacts that you need to produce at this stage include:

- a Test Strategy (overview of testing approach);
- a Test Project Plan (detailed list of testing activities);
- Testing Scenarios (high level test cases for the testers to follow); and
- a Training Plan for the team (when users are new to Workday).

3.1 Roles and Responsibilities

During this early stage of your deployment, the activities are typically performed by the Test Lead, Project Manager, and one or more Test Analysts. Each should be responsible for the following:

Test Manager/Test Lead (referred to as Test Lead for remainder of guide)

- Producing the Test Strategy
- Approving Test Scenarios
- Producing Test Plan
- Preparing other team members for Configuration Review

Project Manager

- Liaising with the Test Manager on scope and timelines
- Working with Test Manager to produce the Test Plan
- Adding key Test Plan activities into the Master Project Plan
- Identifying and assigning appropriate resources for Test Activities

Test Analysts

- Preparing a suite of Test Scenarios.

3.2 Activities and Deliverables

3.2.1 Test Strategy

In this guide, we've outlined the various elements that you should include in your Test Strategy. Your Test Strategy acts as a reference for all team members and stakeholders who need to understand how and when your system will be tested with reference to the Workday deployment plan. Your strategy should take account of your broader technical landscape (integration points, data security) and assess the impact these could have on your planned testing activities.

3.2.2 Test Plan

The Test Plan contains:

- dates for testing activities;
- staffing for the various testing activities;
- environments to be used (i.e. Workday tenants and third-party environments for integrations); and
- key dependencies.

Note: The Test Lead will continue to refine the Test Project Plan as your project continues.

3.2.3 Test Scenarios

Test Scenarios are based on:

- the requirements gathered during the Architect Stage; and
- further input from your SMEs.

For initial testing, these should be high-level scenarios that cover the key configuration requirements and the key processes and activities for each functional area. For example, the scenarios might include an expectation of the outcome of the test or what to check for, but they shouldn't be a step-by-step script (because often testers will follow the script rather than assess if the process actually works for the organisation, and during deployment testing the latter is critical). Later on, your test scenarios will get more detailed, and we talk about this more in Section 6.

Remember to include scenarios for expected results you want to happen as well as those you would not want to occur, e.g. testing the ability for a role to complete an action they shouldn't be authorised to complete. Your Test Lead should prepare and sign off the Test Scenarios before the Configuration Review Stage begins.

3.2.4 Training

For team members who are new to Workday, basic training should be planned before the testing begins.

4 Configuration Review

The aim of the Configuration Review is to allow your Project Team to review the tenant and validate that the design requirements captured during the Architect Stage were communicated correctly and have been accurately applied to your configuration. The test-related activities primarily consist of data validation and unit testing of the configuration.

4.1 Roles and Responsibilities

At this stage, depending on the scale of your deployment, your Test Team may need to expand to comprise the Test Lead, Regional Test Leads, Testers, SMEs, and a Defect Manager. Depending on the size of your deployment and other commitments, your SMEs can perform the roles of Testers. Each role is responsible for the following:

Test Lead

- Ensuring that all scenarios have been prioritised so that high-risk, high-priority scenarios are tested before the lower risk ones
- Ensuring team has been assigned scenarios for each functional and technical area that is in scope for the Configuration Review
- Ensuring the Configuration Review schedule is adhered to and that the required reporting and tracking is distributed during the review phase
- Providing guidance to team members on the process for creating the test scenarios and also ensuring that team members have a full understanding of the test strategy for their respective areas
- Producing a Stage Exit Report once the Configuration Review has finished

Regional Test Lead

- Managing test activities for each dispersed location (region/country/market) that they're responsible for
- Reporting progress to the Test Lead
- Liaising with the Defect Manager to triage issues raised

Testers

- Executing the test scenarios
- Reporting any issues they identify
- Retesting scenarios once the issues have been addressed by the Configuration Team

Defect Manager

- Triaging issues raised by the Test Team, ensuring that the issues are documented clearly, are not duplicated, and are valid
- Assigning issues to the relevant functional consultant within your Configuration Team (see Glossary for Team definition)
- Monitoring the issues logged in your defect tracking system and providing issue reports to the Test Lead on a daily basis.

It may be that your Test Lead also performs the role of the Defect Manager within the project.

4.2 Configuration Review Entry Criteria

For your test event to run smoothly, make sure to meet the following Entry Criteria before beginning the Configuration Review:

- The Test Lead has completed your Test Project Plan, detailing the staging and timelines for the Configuration Review.
- Your Test Lead has prepared and approved your Test Scenarios.
- Your Configuration Team has performed a successful Smoke Test on the tenant. (See Glossary for definition of smoke test).

4.3 Activities

During the Configuration Review Stage, your Test Lead divides the Test Scenarios among the Test Team, often by team responsibility and/or region. The team then executes their assigned test scenarios. The testers raise any issues that they find on the defect tracking system, which are in turn assigned to the relevant Defect Manager. The Defect Manager then triages the issues and assigns them to a member of the Configuration Team.

If Testers report a high number of issues, or the testing has resulted in the need for a high degree of change to your tenant's configuration, you'll need to complete more than one review cycle and plan this accordingly. When testing is finished and the Configuration Review is complete, your Defect Manager should complete a Configuration Review Report to provide a summary of the defects encountered and the status of these defects.

4.3.1 Organisational Set Up Verification

Your Testers need to carry out the following tests on every organisation type in your system:

- Verify all organisations are accurate, including any superior or subordinate organisations (i.e. the values reflect what was provided for the data transformation process).
- Verify that the organisation types and sub-types are accurate (see below).
- Verify that you have assigned security groups to the correct individuals (user based) and assigned individuals to role-based security groups on the correct organisations.

In addition, for each specific organisation type below, Testers must carry out the following checks to ensure accuracy:

- **Supervisory Organisation**
 - Start at the top of the organisation and ensure that the top levels are correct (e.g. the CEO and his direct reports, the Executive Directors and their direct reports, and so on).
 - Ensure that the Organisation Viewer is used to verify the structure of the organisation (i.e. not just looking at a person's manager).
 - Verify that the Organisation Restrictions and Defaults are correct.
- **Locations**
 - Verify that locations and location hierarchies are set up and appear as stated in the requirements.
- **Companies**
 - Check that all required and defined companies are set up and appear as stated in the requirements (e.g. verify that the Tax IDs have been set up correctly).
- **Cost Centres**
 - Check that all cost centres and cost-centre hierarchies are set up and aligned with the requirements. For each cost centre, check that the correct roles are assigned.
- **Regions**
 - Verify that all regions and region hierarchies are configured as detailed in the requirements.
- **Business Units**
 - Verify that all of the business units and unit hierarchies are accurate.
- **Matrix**
 - Check that all of the matrix organisation assignments are accurate and in line with requirements.
- **Custom Organisations**
 - Ensure the values, security roles assigned to the custom organisations, and hierarchies have been configured as documented.

4.3.2 Business Process Unit Test

Testers execute the business processes that have been configured on the tenant at this stage. The purpose of the unit test is to ensure that:

- the flow of the process is correct;
- appropriate security groups can initiate the business process;
- eligibility rules are in place and transacting accurately;
- any associated notifications are routed correctly; and
- notification details are correct.

Your testers also need to execute key actions that are available to users on business processes (e.g. cancel, reject, rescind) to ensure that the impact of inflight business processes are understood.

4.3.3 Report/Audit Review

Reports and audits are available on the tenant at this stage. These audit reports can be helpful in verifying that your tenant configuration is correct (e.g. roles are assigned at organisations correctly). Review the audit reports to ensure that you identify any issues, and use them to help you with data validation activity on your tenant.

4.4 Configuration Review Exit Criteria

Before the Configuration Review is complete, make sure to meet these Exit Criteria to ensure you've been thorough and to provide confidence before starting the next stage:

- The Test Team has completed all high- and medium-risk Configuration Review test scenarios, and in an ideal situation has completed the low-risk scenarios as well.
- Your Testers have updated the defect tracking system with all issues that they identified.
- The Configuration Team has reviewed and addressed all critical or high-risk/high-severity defects (or, alternatively, the Project Management team has provided signoff to proceed).
- The Test Lead has completed and distributed the Stage Exit Report, which will include the Configuration Review report prepared by the Defect Manager (mentioned above), along with the test progress summary.
- The Test Lead has held a meeting with the Project Team to go over the Stage Exit Report and any outstanding items.

5 The Test Stage

The Test Stage is a second round of testing of your configuration. You carry out this testing against a more complete tenant build with additional data migrated to the tenant. As a result, this phase tends to be longer and includes a broader set of representatives from your business beyond the core Project Team. The purpose of the Test Stage is to comprehensively test your configuration of the system, including:

- Organisational Set Up
- Business Processes—unit testing any updated configuration and end-to-end transactional testing (including related integration points)
- Integrations—including testing upstream and downstream system impact
- Security.

There are several facets to the Test phase with multiple parties involved, so it's important that you work with your deployment partner to align the tenant strategy and test plan in advance, prior to communicating to your Project Team (see section 8).

5.1 Test Case Preparation

After the Configuration Review, you'll need to prepare test cases for the Test Stage. Use the test scenarios from your Configuration Review as a starting point, and then build on them to provide a more detailed description of what should be tested. Also, extend the scope of coverage to ensure that each detailed requirement and potential scenario is considered. During the Configuration Review phase of testing, your testers are more conversant with design decisions that you have made, and therefore able to work off high-level scenarios. In contrast, during this phase of testing you'll engage users who are less familiar with design decisions and who may not have had much exposure to the Workday product. Take care to provide them with more detail on the test scripts so that they can navigate successfully.

5.2 Roles and Responsibilities

At this stage, the Test Team typically comprises the Test Lead, Regional Test Leads, Testers, SMEs and a Defect Manager. Each role is responsible for the following:

Test Lead

- Providing signoff that the tenant is acceptable to begin the testing
- Overseeing the testing activities and addressing any logistical concerns that may arise
- Collating the testing results on a daily basis and escalating any critical issues to the relevant stakeholders

5 The Test Stage

- Running regular status updates and reporting back to the Project Leadership Team on testing progress
- Providing the final signoff for the exit of the stage

Regional Test Leads

- Managing test activities for each location (region/country/market) that they're responsible for
- Reporting progress to the Test Lead
- Liaising with the Defect Manager to triage issues raised by their team

Testers

- Executing the test scenarios
- Reporting any issues they identify
- Retesting scenarios once the issues have been addressed by the Configuration Team

Defect Manager

- Triageing issues raised by the Test Team, ensuring that the issues are documented clearly, are not duplicated, and are valid
- Assigning issues to the relevant Project Team member
- Monitoring the issue in your defect tracking system and providing issue reports to the Test Lead on a daily basis.

5.3 The Test Stage Entry Criteria

For your test event to run smoothly, make sure to meet the following Entry Criteria before beginning the Test Stage:

- The Test Lead has completed your Test Plan, detailing the staging and timelines for the Test Stage activities.
- The Test Lead has prepared and/or approved all test cases
- The Configuration Team has performed a successful Smoke Test on the tenant.
- All necessary third-party testing environments are available.
- Support is available from any third parties required for Integration testing.
- The Test Lead has defined issue management procedures and agreed these with the Project Team.

5.4 Functional Testing

Functional testing activities during the Test Stage focus on the testing of the business processes that have been set up for each functional area: Core HCM, Absence, Time Tracking, Compensation, Benefits, etc.

Your Testers need to test each business process (BP) to verify that it's executing as per the specification:

- Roles that are assigned in BPs are in line with expectations in terms of initiators and approvers.
- There are no unassigned roles specified within the BP.
- Routings, rules, thresholds, and any country-specific configuration are set up correctly.
- Negative testing—all validation rules are configured correctly and appropriate warning or error messages are displayed (e.g. one-time payment of \$10M should cause an error).
- All notifications and help texts are correct.
- In addition, we recommend that the functional testing activities also cover the following:
 - For each Calculated Field, validate that they are working as specified.
 - For each Worklet, ensure that the correct items appear and that the required roles only have access to each Worklet.
 - Where Quicklinks are configured, check that the target link is valid and accessible, especially when the link is to an external endpoint.
 - Validate that all the functions that are intended for Mobile access are verified from each in-scope device type (e.g. iPhone, Android).

5.5 Integration Testing

The purpose of integration testing is to ensure that the data being input by and output to third party systems is working in line with the design specifications. These testing activities require the support of your deployment partner, system vendor and your IT team.

5.5.1 Outbound Integrations

For each outbound integration, there are two primary test functions that need to be carried out:

- tests that all appropriate transactions are picked up by the integration; and
- tests to verify that the fields' outputs are mapped correctly.

For all fields in each integration, ensure that all data is correctly exchanged. Pay particular attention to fields that require data mapping (e.g. Gender field is mapping 'Female' to 'F' and 'Male' to 'M') to ensure mapping tables are working as stated in the requirements.

5.5.2 Inbound Integrations

Each of your internal integration owners need to carry out these checks to validate all data mappings for your inbound integrations. In addition, it's likely that inbound integrations will launch business processes automatically (e.g. executes Hire) and auto-populate values in several BP steps. Therefore, it's important for each of your internal integration system owners to check that the BP executes correctly and that the pre-populated values appear as expected in each integrated system. It's also important to check the import for all countries, as each of them will have different formats for data such as names, addresses, and government IDs.

5.5.3 Correct Transaction Testing

When testing integrations, it's critical to execute all business processes and Workday tasks that will impact the integration in Production. This is to ensure that all of the business scenarios that are carried out on your system are picked up correctly by the integration. The types of transactions vary by integration type. Here are some examples of the most common transactions and integration types:

For Payroll integrations, run all business processes and tasks affecting worker payment calculation and payslip generation. Also ensure that:

- the output integration is loaded to your third-party payroll system; and
- you execute a payroll to ensure that the process works end to end.

If you're deploying Workday Financials, also include a GL integration to ensure that the payroll scenarios detailed below result in correct postings to your GL.

Example scenarios include:

- compensation change
- job change (promotion, demotion, lateral move, transfers)
- hire
- re-hire
- termination
- contact details change

- name change
- one-time payments
- recurring earnings and deductions (assign and remove)
- placing worker on long-term leave of absence (e.g. sabbatical, maternity)
- granting worker short-term time off (e.g. sick leave, compassionate leave)
- returning worker from leave.

For Recruiting Outbound Integrations (job requisitions), run all business processes and tasks that might affect offer generation, including:

- creating job requisition
- changing job requisition
- closing or freezing job requisition
- hiring worker into open requisition.

For Absence Outbound Integrations, run all business processes (or tasks) affecting absence data, for example:

- placing worker on long-term leave of absence (e.g. sabbatical, maternity)
- granting worker short-term time off (e.g. sick leave, compassionate leave)
- time off corrections
- returning worker from leave.

For Benefit Integrations, run all business processes (or tasks) related to and relevant to worker benefit plans, for example:

- enrolling worker into Benefit Plan
- making worker eligible for Benefit Plan
- changing worker data so that worker is no longer eligible for Benefit Plan
- triggering Open Enrolment benefit event and electing or waiving Benefit Plans on worker's behalf.

Executing the scenarios above helps to ensure that you thoroughly test the business process impact on integrations and successfully transmit files to the vendor.

5.5.4 Additional Integration Tests

If your team has the resources available, we recommend that you carry out these additional integration tests to ensure thorough test coverage. Note that these additional tests require an element of understanding the integration system for your internal team to be able to take ownership of the integration once you move to production.

Fault scenarios

- ensuring that data is properly validated and any missing or incorrect data is being flagged (via integration messages)
- ensuring that notifications are being dispatched as designed.

Integration business logic

- ensuring that all business logic described in integration documentation works as designed.

Effective date change detection

- ensuring that the integration sends the data according to the effective date specification (i.e. will or will not send events effective in the future, etc.).

5.6 Security Testing Activities

The purpose of Security testing is to ensure that:

- your system's security configuration is correct for both role-based security and user-based security; and
- each user's access is in line with your specified requirements.

These tests should also verify that the data each user and/or role type has access to view or edit is correct. For example, your Testers should ensure that managers only have access to sensitive worker data in their direct reporting hierarchy.

Your security tests need to validate that users:

- have access to the appropriate related actions, tasks, reports and data fields;
- only have access to the organisations for which they're responsible; and
- have appropriate access (if authorised) to run the integration, change the integration, and view integration results (integration events).

5.7 End-to-End Testing Activities

End-to-end testing is a full lifecycle test event. One example is hire to terminate, where you test the acceptance of a new starter from the recruitment system via an integration...through the onboarding business process and various BP changes...and finally the termination of the worker. The purpose of this type of test is to ensure that the full chain of processes work together as a cycle and that approvals are appropriate. Integration points are also tested as a result of this end-to-end approach.

Per Workday's methodology, plan and arrange end-to-end testing activities as a single testing event that draws in the various teams that are responsible for each functional area and each integration. It's critical to organise your end-to-end activities within the overall project plan that's been agreed with your deployment partner. You will also need to engage with the broader business users in both this end to end testing and User Acceptance Testing (UAT), and in order to get alignment with third-party vendors in terms of integration testing. Success of this activity is very much reliant on early and clear planning, engaging all stakeholders and ensuring a controlled UAT activity.

In terms of proxy access, consider conducting a portion of your end-to-end activity (especially UAT) with Proxy set to 'off' so that your team gets experience of performing the roles on processes that they will use in the production environment.

5.8 The Test Stage Exit Criteria

Make sure to meet the following Exit Criteria before completing the Test Stage:

- The Test Team has completed all high- and medium-risk Test scenarios, and in an ideal situation has completed the low-risk scenarios as well.
- All in-scope integrations and custom reports have been tested, with endpoints and testing through to third-party systems completed.
- Your Testers have updated the defect tracking system with all issues that they identified.
- The Configuration Team has reviewed and addressed all critical or high-risk/high-severity defects (or, alternatively, the Project Management team has provided signoff to proceed).
- Where there are critical or high defects that have not been resolved, workarounds have been put in place.
- A prioritised list of changes has been created that may require early configuration once your production tenant is available.
- The Test Lead has completed and distributed the Stage Exit Report, which includes the defect summary report mentioned above along with the test progress summary. A key addition to the Stage Exit Report for this phase is the status of testing each of the in-scope integrations.
- The Test Lead has held a meeting with the Acceptance Test Team to go over the Stage Exit Report and any outstanding items.

6 User Acceptance Testing

The purpose of User Acceptance Testing (UAT) is to validate from an end user's perspective that your system is fit for purpose. As such, it's more of a review of the processes built and a period of activity to enable exposure to the Workday system to the broader business. At this point the business processes have already been through two rounds of thorough testing (Configuration Review Testing and End-to-End testing), so most defects and issues should be resolved.

UAT requires engagement from individuals from key areas of the business to review and confirm scenarios that relate to their typical daily interaction with the system. The User Acceptance Test Team is responsible for:

- completing UAT scenarios as delivered; and
- giving feedback on any critical issues on completion of scenarios.

6.1 Roles and Responsibilities

The Test Team for this stage comprises the Test Lead, Regional Test Leads, the Defect Manager, and your assigned end users. Each role is responsible for the following:

Test Lead

- Working with the Regional Test Lead to ensure that people have been assigned to the preparation tasks for each functional and technical area in scope for the UAT phase
- Managing the logistics for each UAT session
- Collating the testing results each day and escalating any critical issues to the relevant stakeholders
- Running regular status updates and reporting back to the Project Leadership Team on testing progress
- When the testing stage is complete, reporting on UAT activity as part of the Stage Exit Report

Regional Test Leads

- Managing test activities for each dispersed location (region/country/market) that they're responsible for
- Reporting progress to the Test Lead
- Logging any issues to the defect tracking system as raised by the User Acceptance Test Team

Defect Manager

- Triaging issues raised by the User Acceptance Test Team, ensuring that the issues are documented clearly, are not duplicates, and are valid
- Assigning issues to the relevant Project Team member

- Monitoring the issue in your defect tracking system and providing issue reports to the Test Lead on a daily basis.

6.2 UAT Entry Criteria

For your test event to run smoothly, make sure to meet the following Entry Criteria before beginning UAT activities:

- The Project Steering Group and Regional Test Leads have agreed the UAT plan.
- The Configuration Team has performed a successful Smoke Test on your tenant.
- The Test Lead has reviewed and approved the Test Scenarios.
- The Configuration Team is available to provide support during the UAT testing stage.
- All UAT users have been made available for the testing activities.
- The Test Lead has defined issue management procedures and agreed these with the Project Team.

6.3 UAT Testing Activities

The core activity during the UAT stage is the hosting of UAT sessions, which enable the end users to engage in a meaningful way with your Workday system. These sessions tend to be different than the formal testing that has already taken place; the tests are typically a defined list of scenarios that your users execute, and these tend to have a more scripted flow.

When organising these sessions, consider providing 'model offices' that your users can drop into at their location to execute their defined scenarios. This allows appropriate functional support to be on hand to assist with any questions that users have. A typical UAT session looks like this:

- Introduction: The Test Lead (and/or Regional Test Leads) introduce the UAT principles and provide a brief overview of the Workday system.
- Live Session: The User Acceptance Test Team accesses your Workday system. They follow a number of scenarios relevant to their role in the company in order to validate the system is fit for purpose.
- Collation: The Test Lead (and/or Regional Test Leads) collect the feedback or issues raised by the user groups and record these in the defect tracking system. Note: Some Test Teams structure their UAT to include training as part of the session.

6.4 UAT Exit Criteria

Before the UAT stage is complete, make sure to meet these Exit Criteria:

- The testers in the User Acceptance Test Team have executed all UAT scenarios.
- The Regional Test Lead has updated the defect tracking system with all identified issues.
- The Configuration Team has successfully reviewed and addressed all critical or high-level severity defects (or, alternatively, the Project Management team has provided signoff to proceed).
- The Test lead has completed and distributed the Stage Exit Report, which includes a list of defects encountered and the status of each.
- The Test Lead has held a meeting with the User Acceptance Test Team to go over the Stage Exit Report and any outstanding items.

7 Payroll Parallel Testing

The aim of Parallel Testing is to ensure that the payroll processes and integration(s) configured in and with Workday provide the same outputs as your existing payroll process. It's important that the existing process has a high level of accuracy, otherwise there will be significant differences between the Workday payroll calculations and your legacy system, and these differences will make it difficult to identify the root cause in any calculation difference.

You should also carry out parallel testing when you deploy Workday Payroll as part of your overall engagement.

7.1 Roles and Responsibilities

At this stage, the Test Team typically comprises the Test Lead, Payroll Managers, HCM SMEs, and Finance SMEs. Each role should be responsible for the following:

Test Lead

- Ensuring that the teams above have been assigned to the preparation tasks for each activity in scope for the Parallel Testing

HCM SME

- Executing HCM transactions that are inputs to parallel run (e.g. new hires, change jobs, compensation change activity) within agreed timeline

Payroll Managers

- Validating that data is aligned to end of previous pay cycle, prior to executing parallel run
- Entering pay-specific data to the parallel tenant to ensure mirroring of legacy payroll cycle transactions
- Validating that the system meets their typical business function requirements
- Validating that the results obtained from Workday match the legacy system

Payroll Vendors

- Providing testing environments for their payroll systems
- Participating in the payroll run reviews process

Finance SMEs

- Reviewing the resulting General Ledger postings from the payroll parallel.

7.2 Payroll Parallel Testing Entry Criteria

Before you begin Parallel Testing activities, make sure to meet the following Entry Criteria:

- The Test Manager has completed your Parallel Test Plan, detailing the staging and timelines. This should include a decision on which pay periods are going to be used for the parallel run process.
- The Test Lead has prepared and approved the Parallel Test Scenarios.
- Your Configuration and Technical Teams have performed a successful Smoke Test on the tenant.
- All necessary payroll vendor testing environments are available.
- UAT has been completed and signed off and there are no ongoing configuration changes in progress from UAT or other previous test cycles, as these changes will impact pay calculations.
- There is a level of stability in the legacy process that enables a comparison of the Workday payroll configuration, analysis of issues, and appropriate action.
- Defects from testing which will have an impact on payroll process or calculation have been resolved prior to the parallel run. Otherwise, there will be issues in the calculated results on Workday.

7.3 Payroll Parallel Testing Activities

Parallel testing is the process of running the exact same tests on different systems that contain identical data. With all variables identical except for the systems themselves, any differences between your current application's output files and Workday's output files will tell you where the issues lie.

The key to reliable test results is consistent data maintenance. Any transaction on your live system that's pertinent to payroll needs to be dual keyed or 'maintained' in the Workday testing environment in order to keep the two systems in sync and allow you to compare like-for-like test results. This is a time-consuming activity that you need to plan for.

The Payroll Manager runs the existing payroll process and then uses the same data to run the new Workday integrated process in parallel. The Payroll Manager compares outputs from each to identify any discrepancies. You may need additional files or data from 'offline' or outside of the automated process, where data is provided from additional sources as an input to the payroll process (e.g. benefit election changes from benefit provider to load to Payroll)

HCM SMEs and the Payroll Manager enter the data that was processed in your live system into the Workday parallel tenant. When the Workday payroll has been processed, the Payroll Manager compares the two payroll runs for any differences. Where any are found, the Payroll Manager reviews the data and steps back through the process to see why the results are different.

Per Workday's methodology, we strongly recommended that you complete at least two periods of parallel runs for each payroll before moving into production.

As part of planning, consider which pay periods are appropriate to run in parallel—do not select pay periods that have a significantly higher transaction volume (e.g. seasonal hires) or a lower volume of transactions (holiday periods).

7.4 Sample Payroll Parallel Test Plan—Steps Through Testing

1. Clone your Workday tenant

Once your Workday End-to-End tenant is available, request a copy of the tenant for Parallel Run 1 activities. This needs to contain the worker data that reflects the position at the date of the completion of the pay cycle prior to the parallel run period. It may be that as configuration changes are made through testing, the parallel tenant will need to be updated to keep it line with latest configuration.

2. Your Payroll Manager and Test Lead selects the period for the parallel run

Based on comments above, there are a number of criteria for selecting the appropriate period for a parallel cycle.

3. Your payroll vendor takes a copy of their system

The date will depend on the Parallel Test Period (See section 7.3).

4. Track changes in the legacy HR system (spreadsheets/templates)

Consider what else needs to be included in the payroll runs (e.g. previous period overtime amounts)—all transactions from the legacy pay run should be replicated into the parallel tenant. This should be owned by your Payroll Manager.

5. Input changes into your Workday parallel tenant

The HCM SMEs and Payroll team replicate all transactions that have been processed on the legacy payroll system. The effort for this can be significant if you need to make a lot of updates, but you can expedite the process by adapting BPs or using iLoad and EIBs, etc. Also make sure to consider retro-active changes, as these could cause issues when loading bulk data.

6. The Project Team manually checks the outputs

This ensures that all required transactions have been executed against the Workday tenant.

7. The Payroll Manager compares the Workday Payroll to the Legacy calculations

There are a number of ways of achieving this. A simple Excel extract comparing Gross and Net Pay values will enable a view of what differences there are within the pay calculation. There are also tools delivered by Workday (Payfact and Prove) that provide more detailed analysis of variances at a pay component level.

7.5 Payroll Parallel Testing Exit Criteria

Make sure to meet the following criteria before completing the Parallel Test Stage:

- The tolerance levels for discrepancies are within the limits outlined in the Parallel Testing Plan.
- The Test Lead has completed and distributed the Stage Exit Report, which includes a list of all defects encountered and the status of each.
- The Test Lead has held a meeting with the Acceptance Test Team to go over the Stage Exit Report and any outstanding items.

8 Tenant Strategy

During your Workday deployment, a number of Workday tenants will be available to your Project Team. Your Project Manager must prepare a detailed Tenant Management Plan/Strategy that outlines the purpose and activities of each tenant at each stage of the deployment. Examples of tenant uses include:

- Master Tenant—a copy of a delivered prototype that contains the master configuration during testing and other activities
- Integrations Tenant—a copy of a delivered prototype that is available to the Integrations Team for their configuration activities
- Training Tenant—a copy of a delivered prototype that is made available to the Training Team
- Testing Tenant—a prototype tenant that is made available to the Testing Team to perform testing activities
- Parallel Tenant—a copy of a tenant with the latest configuration and agreed data set that is used for testing Workday Payroll and comparing to your legacy system.

Each tenant is refreshed at various milestones during the project. It's important to plan ahead to ensure that any test data that will be required during the testing stages is prepared before each test window. On each of the tenants, you should also consider how you are going to use proxy to assist with the testing activities.

Furthermore, in cases where testers shouldn't have access to sensitive data, that test data needs to be anonymised. If testers access sensitive worker data in countries outside of their location, consider whether specific data protection rules need to be adhered to. This is especially important where Workers' Councils are applicable.

A Tenant Strategy also highlights which third-party system environments will be required during the deployment for testing at each stage.

9 Defect Management

During each testing stage, issues are raised by the Testers. To ensure consistency and avoid confusion, clearly define the process for how to raise defects, what the process is for fixing and retesting issues, document these processes, and distribute the formalised process to the Testers and Project Team.

9.1 Defect Lifecycle

When Testers find issues during their testing activities, they log these on your defect tracking system (usually a central desktop or similar project-collaboration portal). The illustration below outlines the cycle of the defect once it has been raised.

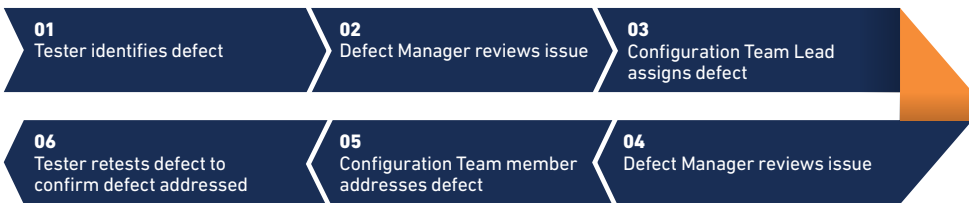


Figure 2—Issue Lifecycle

1. Tester identifies a defect and logs it in the defect tracking system. The defect is assigned to the Defect Manager by the Tester as he or she logs the issue.
2. The Defect Manager reviews new defect for clarity and for duplication. If there are questions with the details of the defect, the defect may be sent back to Tester for further details or elaboration on the defect description. If the defect already exists or is determined to be invalid, the Defect Manager rejects the defect, updates the resolution accordingly, and adds comments. Otherwise, the Defect Manager ensures that the defect severity and category are correct and assigns it to the Configuration Team Lead.
3. The Configuration Team Lead assess the defect. If the defect is deemed not to require a fix, he/she assigns it back to the Test Lead to reassess (who either assigns the defect back to Lead Consultant or rejects the defect). If the defect is deemed valid, it's assigned to the Configuration Team member responsible for the affected area.
4. The Configuration Team member investigates the defect based on the severity that has been assigned. If he/she needs further clarification on it, he/she contacts the Tester who reported it. Once the Configuration Team member has applied

the fix to the testing tenant, he/she reassigns the defect back to the Tester who initially raised it and updates the comments section of the defect with a detailed explanation on the resolution and specific details of changes made.

5. If the fix fails review, the Tester reassigns the defect back to the Configuration Team member for further work to resolve the defect. If the defect passes review, the Tester updates the status of the defect and assigns it to the Configuration Team Lead (for further consideration should the fix need to be applied to other tenants).

9.2 Defect Severity Criteria

The Defect Severity level refers to the potential impact of the defect on the product or organisation.

Severity	Description
Critical	The defect results in the failure of the complete system or a functional area within the system, or results in the failure of a critical service provided by the system to external parties.
High	The defect results in the failure of the complete system or a functional area within the system, or results in the failure of a critical service provided by the system to external parties. There's no way to make the failed component(s) work; however, there are acceptable processing alternatives that will yield the desired result.
Medium	The defect does not result in a failure but causes the system to produce incorrect, incomplete, or inconsistent results, or the defect impairs the system's usability.
Low	The defect does not result in a failure, does not impair usability, and the desired processing results are easily obtained by working around the defect. Additionally, the issue is the result of non-conformance to a standard or is related to the aesthetics of the system.

9.3 Defect Priority Criteria

The Defect Priority level refers to the level of impact the defect may have on testing.

Severity	Description
High	Severely impacts testing. Testing is affected and there is a significant impact—an area of testing is unavailable.
Medium	Medium impact on testing. Only a small area is impacted. Testing can continue but won't be complete until the defect is resolved. A workaround is available.
Low	Minimal impact to testing, testing can continue, e.g. spelling mistake.

10 Test Reporting

During each testing stage of the deployment, the Test Manager produces and distributes status reports. Example reports include:

Daily Progress Reports

- This report provides an update on the progress made each day during the testing stage.
- It highlights a number of testing statistics, including:
 - the number of test cases run compared to the target number of test cases planned;
 - the number of defects raised;
 - a breakdown of defects raised by severity; and
 - a breakdown of defects raised by geography.
- It describes any issues with testing, i.e. any blockers for the Test Team, along with any alternate instructions.
- The report is shared with the core Project Team.

Weekly Status Reports

- The weekly status report provides an update on the testing progress made each week during the testing stage.
- It includes a RAG status on the following key items, including:
 - number of defects found per functional area;
 - number of defects found per geography; and
 - progress of the testing activity per plan.
- It includes a brief overview of any high-severity defects raised.
- The report is reviewed with the Project Manager before it's distributed.
- It is sent to the Steering Group and core Project Team.

Testing Stage Exit Report

- The main purpose of the report is to provide information to the Project Team in order to decide whether or not the exit criteria have been met and the team is happy to proceed to the next stage of the project.
- The report includes an overview of the activities performed by the Test Team during the stage and compares that with the planned activity set out in the Test Project Plan.
- It includes a detailed list of all of the defects raised and the latest status of each.
- It clearly highlights any defects that are still open.
- It is shared with the Steering Group and the core Project Team.

11 Automated Testing

Given the complexity of Workday deployments and the amount of effort required to thoroughly test all of the various components, more and more clients are looking to automated testing solutions to help reduce the workload. Kainos uses its industry leading automation product Kainos Smart, the only testing product specifically designed for Workday deployments. This powerful and easy-to-use tool encompasses business processes, security and integrations for Workday HCM, Workday Recruitment, Workday Financials, and Workday Payroll. The table below details how Kainos Smart can improve the coverage and quality of testing for each of these areas.

Business Process Testing

The Issue	How Kainos Smart Helps	Benefit
There are typically 100s of individual test cases required to test a single business process for a Workday deployment. Each individual test can take anywhere from 15 minutes to 1 hour to complete. Given the multiple cycles of testing involved, the amount of manual effort required is very large—often tying up SMEs for extended periods.	Smart can test business processes automatically via a single button click. Smart can also test multiple business processes simultaneously. Test cases are very simple to create in Smart and once set up can be re-executed over and over with no additional overhead.	More comprehensive test coverage. Detailed audit trail (i.e. number of tests, dates, completed by, success %). Freeing up of valuable subject matter experts.

Integration Testing

The Issue	How Kainos Smart Helps	Benefit
<p>Testing integrations involves checking to see if business transactions entered on Workday cascade successfully into downstream systems. This means that the Integration Test Team has a huge dependency on their Business Team, as they rely on them to execute the business transactions.</p> <p>The net result is often significant delays in completing integration testing.</p>	<p>The integration Test Team can use Smart to execute the business processes, ensuring they have the necessary transactions in the system to complete their integration testing.</p> <p>This removes the dependency that the Integration Team has on the Business Team, thus allowing integration and business process testing to run in parallel.</p>	<p>Ability to commence integration testing earlier in the test window.</p> <p>Ability to populate data required on the system for integration testing without manual effort or SME involvement.</p>

Security Testing

The Issue	How Kainos Smart Helps	Benefit
<p>Ensuring that your security configuration is accurate is one of the most important aspects of a company's Workday deployment.</p> <p>The manual effort required to verify this, however, can be substantial and extremely laborious. Tens of thousands of individual checks are required to achieve adequate coverage and risk mitigation.</p>	<p>Smart allows users to set up a very comprehensive security test suite in a matter of days. This test suite can then be executed automatically.</p> <p>Smart verifies that each user has permissions for all the actions that they should have. Crucially, it also checks that they don't have the permissions for those orgs/workers/ actions that they shouldn't have access to.</p>	<p>Ability to commence integration testing earlier in the test window.</p> <p>Ability to populate data required on the system for integration testing without manual effort or SME involvement.</p>

Another crucial benefit of using Smart during deployment is that on 'go live', you have a ready-made automated regression pack that you can use again to regression test your Workday deployment as you move to life in production—for weekly Workday updates, Workday's major bi-annual releases, and as needed for your own ongoing configuration changes.

Smart is currently being used successfully by over 140 Workday customers. If you'd like further information about the platform and how it can ease your testing, please contact us using the details on the back cover of this guide.

12 Glossary of Terms

Term	Description
Configuration Review	The Configuration Review is the first Testing Stage where the Project Team reviews, at a high level, the initial prototype provided.
Configuration Team	The deployment team that is responsible for maintaining the key elements of your configuration. This will almost always be your deployment partner.
End-to-End Testing	E2E testing covers the full flow of end-to-end processes between multiple product functions and integrations.
Parallel Testing	Parallel testing ensures that processes configured in Workday meet the same results found in your current legacy application. This is applicable if you are deploying a Payroll integration or Workday Payroll.
Smoke Test	When a tenant build is complete, the build teams should perform a pre-defined shortlist of tests to ensure the core functionality is working as expected before it's released for a test cycle.
Tenant	A Workday tenant is defined as an instance of the Workday software. During a rollout, there will be multiple tenants used for various purposes, e.g. core configuration, integration configuration, training etc.
Test Case	A Test Case is a more specific and detailed scenario that a tester should follow in order to validate that the Workday Configuration is set up as required.
Test Plan	The Test Plan is the detailed schedule of testing activities together with the responsible owner for each, and it's often integrated into the overall project plan.
Test Scenario	A Test Scenario details, at a high level, a specific scenario that a tester should follow in order to validate that the Workday Configuration is set up as required.
Test Strategy	The Test Strategy details the testing approach and activities for the Workday rollout.
User Acceptance Testing	UAT is testing that allows a select group of end users to accept the Workday system before it goes live based on their testing of the software.

kainos® WorkSmart

European Headquarters

4-6 Upper Crescent

Belfast

County Antrim, BT7 1NT

United Kingdom

Phone: +44 28 9057 1100

US Headquarters

4th floor

470 Atlantic Avenue

Boston, Massachusetts 02210

Phone: +1 617 217 2704