CS696F – Computer Science, Spring 2025

Pace University

A blue and yellow logo

AI-generated content may be incorrect.

TEST PLAN DOCUMENT

Xero Accounting Software Testing

Authors: Team 2

Team Manager: Anitha Raj Bale

Date: 4/3/2025

TEST PLAN DOCUMENT

Contents

[Test Plan Identifier 3](#_heading=h.uk9ksccryxs7)

[Introduction (Anitha) 3](#_heading=h.k6xkekt8s21c)

[Test Items (Anitha) 3](#_heading=h.3khu1b1mo3yx)

[Software Risk Issues (Anitha) 4](#_heading=h.rv76lek2m21s)

[Features to be Tested (Harshil) 7](#_heading=h.bb8jiz203r51)

[Feature Not To Be Tested (Harshil) 12](#_heading=h.drgiclcffyqx)

[Item Pass/Fail Criteria (Harshil) 12](#_heading=h.4ev2r4l84sdj)

[Suspension Criteria And Resumption Requirements (Deepak) 1](#_heading=h.7woqp81iguh3)4

[Suspension Criteria 1](#_heading=h.j6n66vacgykl)4

[Resumption Requirements 1](#_heading=h.1mgng1nqnmug)5

[Test Deliverables (Deepak) 1](#_heading=h.d89q6jnafdsq)6

[Remaining Test Tasks (Deepak) 1](#_heading=h.33pi57lqu9tw)7

[Environmental Needs (Moldir) 16](#_heading=h.4c9xw4x5ama2)

[Staffing and Training Needs (Moldir) 17](#_heading=h.ugs0nwrnllqd)

[Responsibilities (Moldir) 17](#_heading=h.s48x07pf5foo)

[Schedule (Rahul) 18](#_heading=h.xnjjw961kit3)

[Risks and Contingencies (Rahul) 18](#_heading=h.iy0nq1llkagb)

[Approvals (Rahul) 18](#_heading=h.db5n4gq1tux7)

# Test Plan Identifier

Xero-ManualTesting-Team2\_TestPlan\_v1.0

# Introduction (Anitha)

This test plan provides a detailed strategy for validating four critical modules of the Xero accounting software:

1. 08. Purchase Orders
2. 09. Expense Claims
3. 10. Bills to Pay
4. 11. Accounts Payable

The goal is to ensure these modules operate accurately, securely, and in line with accounting standards, while addressing concerns such as user permissions, data validation, and system integrations.

# Test Items (Anitha)

* Purchase Orders (PO) Module Test Items::
* PO Creation and Management**:** Testing of the creation, editing, deletion, and approval workflows for purchase orders.
* Supplier Integration**:** Testing of linking POs to suppliers and verifying the accuracy of supplier information.
* PO Status Tracking**:** Testing the tracking of PO statuses (e.g., Draft, Approved, Partially Received, Received).
* Item and Quantity Management**:** Testing the addition, modification, and validation of items and quantities in POs.
* PO to Bill Conversion**:** Testing the conversion of approved POs into Bills to Pay (BTP).
* Multi-Currency Support**:** Testing of currency conversion and exchange rate accuracy for international POs.
* Tax Handling: Testing the correct application of taxes based on PO details and supplier country regulations.
* Data Validation**:** Testing data validation rules to ensure accurate entry of PO details, such as item prices, quantities, and taxes.
* Expense Claims (EC) Module Test Items:
  + Expense Entry and Categorization: Testing of entering and categorizing expense claims (e.g., travel, office supplies).
  + Receipt Upload and Verification:Testing the process for uploading and verifying receipts for expense claims.
  + Approval Workflows: Testing role-based workflows for the approval and rejection of expense claims.
  + Reimbursement Processing: Testing the processing of reimbursements for approved claims.
  + Multi-Currency Support: Testing the accurate handling of expenses in multiple currencies and conversion rates.
  + Expense Limits and Policies: Testing the enforcement of company-specific expense limits and approval policies.
  + Data Validation: Ensuring the accuracy and consistency of expense claim data and categorization.
* Bills to Pay Module Test Items:
  + Invoice  Bill Creation and Management: Testing the creation, editing, deletion, and approval of bills from suppliers.
  + PO Linkage: Testing the automatic or manual linking of bills to purchase orders.
  + Payment Processing: Testing of payment workflows for bills, including partial and full payments.
  + Due Date and Early Payment Discounts: Testing for automatic handling of due dates, early payment discounts, and late payment penalties.
  + Multi-Currency Support: Testing of currency conversion and exchange rates when paying bills in multiple currencies.
  + Tax Handling: Testing the correct application of taxes (e.g., VAT/GST) for each bill.
  + Reconciliation with Accounts Payable**:** Ensuring the bills are properly reconciled with the Accounts Payable module.
  + Data Validation: Ensuring that bill data, including amounts, payment terms, and due dates, are accurately entered and processed.
* Accounts Payable Module Test Items:
* Bill Payment Processing: Testing of the payment processing from accounts payable, including both manual and automated payment methods.
* Vendor Account Management: Testing the addition, modification, and deletion of vendor accounts in the AP system.
* Payment Terms and Schedules: Verifying that payment terms (e.g., Net 30, Net 60) and payment schedules are correctly applied.
* Invoice Matching: Testing the matching of incoming bills or invoices with purchase orders and receipts.
* Multi-Currency Support: Testing of payments made in different currencies and the accuracy of currency conversion.
* Payment Approval Workflow: Testing of role-based workflows for approving payments to vendors.
* Aging Reports: Testing the generation of accounts payable aging reports to track overdue payments.
* Reconciliation with Bank Feeds: Ensuring that payments made through the AP module are reconciled with bank transactions.
* Data Validation: Ensuring the integrity and accuracy of data in the accounts payable process, including amounts, payment terms, and due dates.

# Software Risk Issues (Anitha)

Below are potential risks identified for the Xero Financial Data Entry, Bank Feeds, Invoicing, and Accounts Receivable modules, along with mitigation strategies. Risks are derived from the test scope, cross-cutting concerns, and feature backlogs.

1. Data Integrity Risks

Risk: Invalid or corrupted financial data due to improper field validation (e.g., incorrect item quantities, negative amounts in bills, incorrect tax rates in expense claims).

Impact: Inaccurate financial reports, miscalculation of payments, non-compliance with financial regulations.

Mitigation:

Enforce strict field validation rules (e.g., numerical-only fields, mandatory tax codes).

Test edge cases such as $0 amounts, multi-currency decimals, and negative values.

Validate totals in purchase orders and bills against line-item data.

2. Security and Access Control Risks

Risk: Unauthorized access to sensitive data (e.g., vendor information, expense claims, or payment details) due to misconfigured user roles or weak access controls.

Impact: Data breaches, financial fraud, unauthorized changes to financial records.

Mitigation:

Validate role-based access control (e.g., ensure "Expense Claim Only" users cannot approve or modify purchase orders).

Test entitlements thoroughly for different roles (e.g., ensuring "Read-Only" users cannot delete or modify transactions).

Ensure vendor and payment data are only accessible to authorized personnel

3. Integration Risks

Risk: Failure in integration with external systems (e.g., supplier invoicing systems, bank feeds) or incorrect synchronization of data between modules (e.g., PO to Bills to Pay, Expense Claims to Accounts Payable).

Impact: Discrepancies in financial records, incorrect payment processing, reconciliation errors.

Mitigation:

Test integrations with sample data from external systems, such as supplier invoices and bank feeds.

Validate automated workflows for PO to BTP conversion and EC to AP processing.

Perform end-to-end testing with real and mock data to ensure data flows accurately between modules.

Validate automated transaction matching (5.1 Perform Bank Reconciliation).

4. Compliance Risks

Risk: Tax calculations (e.g., VAT/GST on bills or expense claims) deviate from regulatory standards due to incorrect configuration or software bugs.

Impact: Legal penalties, audit failures, incorrect tax filings.

Mitigation:

Review tax calculation logic, ensuring accurate tax handling across all modules.

Cross-check invoices and bills against tax reports for compliance with local and international tax regulations.

Perform sample tax calculations and validate multi-currency handling.

5. Performance Risks

Risk: System slowdowns during critical operations such as bulk processing of bills, expense claims, or purchase orders.

Impact: User frustration, system timeouts, and delays in financial reporting or payments.

Mitigation:

Load test bulk operations (e.g., bulk expense claim submissions, bulk purchase order generation).

Monitor system performance during high-load scenarios, including processing large volumes of transactions or invoices.

Analyze system response times and optimize database queries for scalability.

6. Concurrency Risks

Risk: Data conflicts when multiple users attempt to edit the same financial record (e.g., editing purchase orders, processing payments, or modifying bills).

Impact: Overwritten changes, data loss, incorrect payment processing.

Mitigation:

Test concurrency scenarios (e.g., two users editing the same purchase order or bill).

Implement mechanisms for locking records or version tracking to prevent conflicts.

Ensure system notifications or warnings are displayed when two users are attempting to edit the same record.

7. Dependency Validation Risks

Risk: Deleting critical data (e.g., deleting a bill linked to a payment or expense claim) could cause broken audit trails or reconciliation failures.

Impact: Loss of transaction history, broken relationships between modules, incomplete financial records.

Mitigation:

Enforce data dependency validation to prevent deletion of records linked to other critical financial transactions (e.g., prevent deleting bills that have been paid).

Implement confirmation prompts and warnings when attempting to delete or modify linked data.

Ensure appropriate handling of dependencies between purchase orders, bills, and payments.

8. Training and Knowledge Gaps

Risk: Testers and users lack sufficient understanding of financial processes, leading to missed defects in critical workflows (e.g., incorrect GL account coding or expense categorization).

Impact: Undetected errors in financial reporting, misallocation of expenses or payments.

Mitigation:

Provide specialized training for testers on accounting workflows, especially for more complex features like tax handling and reconciliation.

Assign Subject Matter Experts (SMEs) for review and validation of complex features.

Use accounting-specific documentation and resources for reference during testing.

9. Environmental Risks

Risk: Resetting of the demo company or test environment during testing, erasing manually entered data (e.g., test purchase orders, expense claims, or payment records).

Impact: Loss of test progress, duplication of efforts, data inconsistencies.

Mitigation:

Schedule tests around the demo company's reset cycle to avoid data loss.

Back up critical test data before performing system resets or environment changes.

Use staging or dedicated testing environments for non-production testing.

10. Third-Party Payment Gateway Risks

Risk: Payment processing failures due to downtime or issues with third-party payment gateways (e.g., Stripe, PayPal) integrated with the system for processing payments.

Impact: Unprocessed payments, delayed bill payments, and revenue loss.

Mitigation:

Test with sandbox environments for payment gateways to simulate real transactions.

Implement fallback mechanisms for payment processing (e.g., manual payment methods or retries).

Monitor the third-party services for uptime and integrate proper error handling for downtime scenarios.

# 

# Approach to System Testing (Harshil Goti)

**Types of System Testing:**

1.    Functional Testing

* Verifies that core functionalities (e.g., 8.03 Create Purchase Order, 10.01 Create Bill, 9.02 Submit Expenses) work as expected against the business requirements.

2.    Integration Testing

* Ensures smooth interaction between modules (e.g., Expense Claims and Payroll Integration in 9.07; Purchase Order flowing to Accounts Payable in 11.01).

3.    Regression Testing

* Confirms that new updates (e.g., automation in 11.03) do not break existing functionality like bill payments (10.02) or vendor creation (8.01).

4.    Security Testing

* Verifies access restrictions for sensitive functions (e.g., admin-only actions in 11.04, restricted PO approval in 8.04).

5.    Usability Testing

* Checks UI/UX efficiency for workflows like expense entry (9.01) or product catalog management (8.02).

6.    Performance Testing

* Evaluates system behavior during high-volume actions like bulk invoice generation (11.01) or recurring bill automation (10.04).

**Testing Techniques:**

* -    Equivalence Partitioning: For validating field inputs like vendor names, bill amounts, tax codes.
* -    Boundary Value Analysis: For testing edge cases (e.g., 0, negative, or extremely high values in 10.01 or 9.03).
* -    State Transition Testing: For workflows like PO approval (8.04) or expense lifecycle from submission to approval (9.05).
* -    Error Guessing: Based on tester’s domain knowledge to catch common financial data entry mistakes or UI inconsistencies.
* -    End-to-End Testing: Ensures complete workflows function seamlessly (e.g., PO → Bill → Payment).

**Tools and Technologies:**

* -    Test Management: qTest, JIRA for tracking test cases, requirements, and defects.
* -    Automation Tools: Selenium for UI automation, Postman for API testing (e.g., payment processing 11.05).
* -    Performance Testing: JMeter or LoadRunner to stress-test bulk features.
* -    Version Control: Git (if applicable to test scripts or integration monitoring).
* -    CI/CD: Jenkins or Azure DevOps pipelines for automated test execution.

# 

# 6) Entry/Exit Criteria (Harshil Goti)

**Entry Criteria:**

* Test environment set up with Xero demo company and required data (e.g., vendor list, product catalog, currencies).
* All functional requirements for modules 08 to 11 are baselined and available.
* Test cases and scenarios are created, reviewed, and approved.
* Integration points (e.g., with payroll, Stripe/PayPal) are available for sandbox testing.
* Access to necessary tools (e.g., qTest, JIRA, automation environments) is confirmed.
* Dependencies like 3rd-party APIs or scheduled data (e.g., recurring bills) are stable.

**Exit Criteria:**

* All planned test cases are executed for modules 08–11.
* All critical and high-severity defects are resolved and retested (e.g., PO creation bugs, payment failures).
* Traceability matrix is updated and covers 100% of functional requirements.
* Test summary report is signed off by stakeholders.
* No unresolved blocker or critical defects remain in production workflows.
* Regression test cycle is complete and passed.
* UAT sign-off (if applicable) is received.

# Suspension Criteria And Resumption Requirements (Deepak)

## Suspension Criteria

Testing will be suspended immediately if any of the following issues are encountered:

1. Failure in Core Functionalities:
   * Creating or processing Purchase Orders (e.g., 8.03 Creating Purchase Order, 8.04 Approving Purchase Order).
   * Managing bills or expenses (e.g., 10.01 Creating Bill, 9.03 Categorizing Expenses)
   * Processing payments (e.g., 10.02 Processing the Payment, 9.04 Consolidate Payments)
   * Malfunction in automation workflows (e.g., 10.04 Automating the Bills for Recurring).
   * Data corruption while managing documents (8.06 Managing Documents) or receipts (9.01 Capturing and Analyzing Receipts).
2. Security and Authorization Breaches:

* Unauthorized access to sensitive admin features (e.g., 11.01 Administer Billing and Invoices, 11.04 Enforce Security and Compliance).
* Incorrectly configured user roles (e.g., standard users being able to approve high-value expenses [9.05] or modify system settings [11.03]).

#### 3. Environmental Instability:

* Disruption or reset of the test environment, resulting in loss of test data (e.g., vendor profiles, purchase orders, expenses).
* Downtime or instability in integrated services (e.g., payroll integration [9.07], currency handling modules [9.06]).

4. Critical Calculation Errors::

* Inaccurate calculations involving:
  + **Multi-currency conversions** (e.g., 9.06 Handle Multiple Currencies).
  + **Expense totals** or **tax components** in billing (e.g., 10.03 Tracking Expenses, 11.02 Oversee Supplier and Cost Management).
* Inconsistent data across financial reports generated from bills or expense modules.

5. **Resource Unavailability:**

* Testers do not have access to required tools or platforms (e.g., JIRA, test environment).
* Test team lacks training on newly implemented features (e.g., 11.03 Implement Automation and System Integrations).

## Resumption Requirements

Testing will resume only after the following steps are completed:

#### **Defect Resolution:**

* All critical defects are resolved, retested, and verified (e.g., PO approval issues [8.04], recurring bill errors [10.04]).
* Fixes are deployed to the test environment, and regression testing confirms no additional impact on core modules.

#### **Security and Authorization Compliance:**

* User roles and entitlements are re-evaluated and corrected (e.g., only authorized personnel can access 11.05 Handle Payment Processing).
* All security issues are audited and confirmed to be resolved.

#### **Environmental Stability:**

* Test environment is restored with clean and complete test data (e.g., vendors, items, and bills).
* Connected systems (e.g., payroll integrations [9.07], external payment systems) are tested and verified via sandbox or mock environments.

#### **Calculation Validation:**

* Currency handling and tax logic are verified against expected business rules.
* Reports and dashboards (e.g., expense summaries, vendor payments) are validated by finance SMEs.

#### **Resource Readiness:**

* Test team has completed training on any updated or newly released features (e.g., 9.02 Submitting and Approving Expenses, 11.02 Cost Management).
* QA Lead confirms all tools, environments, and permissions are in place.

#### **Approvals:**

* QA Lead and project stakeholders sign off on:
  + Critical defect fixes.
  + Security and stability of the environment.
* Updated test cases and plans are communicated to the entire team.

# Test Deliverables (Deepak)

### **1. Planning & Design**

* **FDD & DFDs:** Breakdown and data flows for modules like Vendor Management (8.01), Purchase Orders (8.03–8.05), Expenses (9.01–9.06), Billing & Payments (10.01–11.05).
* **Context Diagram:** Shows interactions with external systems (e.g., payroll, payment gateways).
* **RCT:** Maps core features (e.g., 9.04, 10.05) to system-level concerns (e.g., entitlements, automation).
* **Entitlements Spec:** Defines access for Admin, Finance Manager, etc.
* **User Stories:** Acceptance criteria per role (e.g., "As a Finance Manager, I can approve expenses").
* **Test Roadmap & Plan:** Test cycles scheduled (Apr 25, May 2, May 9); plan approved by QA & stakeholders.

### **2. Test Execution**

* **Test Design & Cases:**
  + *Cycle 1:* Vendor & PO flow (8.01–8.05), Basic expenses
  + *Cycle 2:* Bills, automation, payments (10.01–11.05)
* **Execution Logs:** Run results (qTest), linked defects (JIRA), with screenshots & notes.
* **Defect Log:** Categorized by severity, with resolution status.

### **3. Reporting & Closure**

* **Defect Reports:** JIRA exports with all fix status.
* **Test Summary Report:** Metrics, risks (e.g., role misconfigurations, integration issues), and QA recommendations.
* **Final Presentation:** PowerPoint + Demo covering workflows like PO approval, expense submission, billing, and automation.

# 

# 

# Remaining Test Tasks (Deepak)

### **1.Test Plan Preparation**

* Finalize scope and test cycles covering core features:
  + Vendor/PO (8.01–8.05),
  + Expenses (9.01–9.06),
  + Billing & Payments (10.01–11.05)
* Address cross-cutting concerns like entitlements and automation.

### **2.Test Design & Case Specs**

* Draft 200+ test cases in qTest:
  + E.g., 8.03 Create Purchase Order, 9.05 Approve Expenses, 10.04 Recurring Bills.

### **3.Environment Readiness**

* Seed test data into the environment:
  + 50+ vendors, 100+ POs & bills, multi-currency (USD, EUR, GBP).
  + Roles: Admin, Finance Manager, Accountant.

### **4.Test Execution**

* Execute 3 planned cycles (Apr 25, May 2, May 9) targeting 95%+ pass rate.

### **5.Test Summary Report**

* Report on defect trends, module-wise test velocity, and open risks  
  (e.g., currency issues, automation gaps, role-based access flaws).

# Environmental Needs (Moldir)

**Platform:**

* **Xero Demo Company**

**Test Data Requirements:**

* **Products**: 20+ items/products, priced between $10 and $5,000.
* **Transactions**: 50+ invoices, 30+ bills, 100+ bank transactions, including multi-currency entries (USD, EUR, GBP) with live exchange rates.

**User Roles:**

* **Admin**: Full access to all modules and functions.
* **Sales**: Restricted to invoicing tasks (e.g., creating, editing, and tracking invoices).
* **Read Only**: View-only access to all records without permissions to edit or delete.

**Tools:**

* **qTest** for test case management.
* **JIRA** for defect tracking and issue management.
* **Stripe/PayPal Sandbox Accounts** for payment gateway testing (payment processing and payment failure scenarios).
* **Xero Demo Account** for scenario-specific testing and end-to-end workflows.

# Staffing and Training Needs (Moldir)

**Team Composition:**

* **Testers (4)**: Execute test cases across modules related to purchase orders, expense claims, bills, and accounts payable (e.g., 8.1 – 11.5).
* **Business Analysts (2)**: Validate tax logic, reconciliation workflows, and alignment with accounting standards.
* **QA Manager (1)**: Oversee test cycles, approvals, and risk communication across the Xero modules being tested.

**Training:**

* **Mandatory:**
  + **Xero Basics**: Understanding the creation of invoices, bills, purchase orders, bank reconciliation, and expense claims.
  + **qTest & JIRA Onboarding**: Training for the test case management and defect tracking tools.
* **Optional:**
  + **Advanced Tax Handling (CL-Calculations)**: Detailed training on tax calculations for multi-currency transactions, taxes on bills and invoices.
  + **Workflow Understanding for Modules**: Detailed sessions covering workflows within **Purchase Orders**, **Expense Claims**, **Bills**, and **Accounts Payable**.

# Responsibilities (Moldir)

| **Role** | **Responsibilities** |
| --- | --- |
| QA Manager | - Approve test plans, reports, and defect resolutions.  - Oversee test cycles and deadlines.  - Communicate risks, bottlenecks, and issues to stakeholders. - Validate the final sign-off and manage team resources. |
| QA Analyst | - Design, create, and update test cases in **qTest** based on module-specific requirements (e.g., 8.1 – 11.5).  - Execute test cases - Log defects in **JIRA** with detailed steps and retest resolved issues. - Track defect resolution and analyze test metrics (e.g., pass rate, defect density).  - Focus on cross-cutting concerns like **Field Validation (FV)** and **Security Checks** for user roles and access permissions. |
| Business Analyst | - Define detailed business requirements for test cases, ensuring they align with real-world accounting practices  - Verify alignment of test cases with tax regulations, reconciliation workflows, and other financial processes.  - Assist in **User Acceptance Testing (UAT)** for critical features - Review financial reports (e.g., 8.19, 9.7). |
| Software Tester | - Execute assigned test cases (e.g., testing the creation and approval of purchase orders, payment tracking, and expense categorization). - Log defects in **JIRA**, providing clear steps, expected outcomes, and screenshots to assist in defect resolution. - Retest resolved bugs within 24 hours.  - Populate test data (invoices, bank feeds).  - Collaborate on complex workflows (e.g., 9.10 Reconcile Payments). |

# Schedule (Rahul)

| **Phase** | **Dates** | **Focus** | **Features** | **Story Points** |
| --- | --- | --- | --- | --- |
| **Cycle 1** | Apr 18–25 | Core CRUD | 4.1–4.5, 6.1–6.4 | 25 SP |
| **Cycle 2** | Apr 26–May 02 | Integrations | 5.1–5.12, 7.5–7.9 | 29 SP |
| **Cycle 3** | May 03–09 | High-Effort Tasks | 6.3, 6.5–6.17 | 20 SP |

# Risks and Contingencies (Rahul)

### **Risks and Contingencies**

| Risk | Impact | Mitigation |
| --- | --- | --- |
| Cycle 1 delays due to CRUD failures in Purchase Orders and Expense Claims | Blocks Cycle 2, delaying integration testing. | Extend Cycle 1 by 2 days; reallocate testers to high-priority tasks. |
| Testers lack accounting knowledge in expense approvals and payment processing | Inaccurate test execution and defect reporting. | Assign Business Analysts (BAs) as mentors; provide Xero training resources. |
| Bank feed downtime affecting reconciliation in Accounts Payable | Halts validation of automated transaction matching. | Use pre-imported CSV data to simulate bank transactions manually. |
| Payment gateway failures (Stripe/PayPal) during invoicing and bill payments | Blocks invoicing and payment processing tests. | Switch to sandbox environments for Stripe/PayPal; validate fallback payment methods. |
| Integration issues with payroll and multi-currency handling in Expense Claims | Errors in payroll data and currency conversions. | Conduct early API testing with test data before full system integration. |
| Security misconfigurations in approval workflows for Purchase Orders | Unauthorized users may approve/reject orders. | Validate role-based access permissions before starting approval tests. |
| Recurring bill automation failures in Bills to Pay | Unexpected duplicate payments or missed scheduled payments. | Test automation rules with smaller datasets before full-scale execution. |

# 

# Approvals (Rahul)

**Approval to Proceed:**

* QA Manager: Confirms that testing is complete, defects are resolved, and reports are accurate.
* Professor/Stakeholder: Validates that the outcomes align with academic or business goals.