Test Summary Report

Team 2

Spring 2025  
  
Team Members  
Anitha Raj Bale  
Harshil Goti  
Rahul Choppara  
Moldir Numrakhan  
Deepak Raj Koduru

**Table of Contents**

[1. Purpose (Anitha Raj Bale)](#_9bhkz1q02eyg) 3

[2. Application Overview (Anitha Raj Bale)](#_et0d23q15tg) 3

[3. Testing Scope (Anitha Raj Bale)](#_iaf3b41a8535) 4

[4. Metrics (Harshil Goti)](#_wh3gob3zy21j) 5

[5. Types of testing performed (Harshil Goti)](#_cusfceuabuv2) 8

[6. Test Environment & Tools (Deepak)](#_wyrl9f2wxbvk) 9

[7. Lessons Learnt (Deepak)](#_my2i6myey8rh) 9

[8. Recommendations (Deepak)](#_5xtwdcmbp6p) 9

[9. Best Practices (Moldir)](#_j7tibu2ngir7) 10

[10. Exit Criteria (Moldir)](#_ovt8slr7d1vv) 11

[11. Conclusion/Sign Off (Rahul Choppara)](#_2znzv93xfj5w) 11

[12. Definitions, Acronyms, and Abbreviations (Rahul Choppara)](#_j3ekwiqpwzfa) 11

# 1. Purpose (Anitha Raj Bale)

The purpose of this Test Summary Report is to comprehensively present the outcomes of the testing activities undertaken as part of the software verification process. It aims to provide a clear evaluation of the functional accuracy, system stability, and readiness for deployment of the features tested.

This report specifically documents the testing efforts carried out for the **Purchase Orders**, **Expense Claims**, **Bills to Pay**, and **Accounts Payable** modules within the Xero cloud-based accounting system. These modules were selected based on their critical role in managing financial operations, ensuring compliance with business workflows, and supporting end-to-end accounting functionality. The findings presented herein reflect the results of functional, regression, and integration testing designed to validate that each module operates as intended and integrates seamlessly within the larger system.

Test Summary Report purpose is to summarize the results of the designated testing activities and to provide evaluations based on these results.

This document outlines the activities conducted during testing of the Purchase Order and Expense Claims Modules of the Xero cloud-based accounting system.

# 2. Application Overview (Anitha Raj Bale)

# Xero is a leading cloud-based accounting software platform developed by a New Zealand-based technology company. Designed primarily for small and medium-sized businesses, Xero enables users to manage their financial operations with ease and efficiency through an intuitive, subscription-based model. With a global presence—including offices in New Zealand, Australia, the United Kingdom, and the United States—Xero’s platform is trusted by organizations across more than 180 countries.

# At the core of Xero's functionality is a single unified ledger, which allows multiple users to access and work collaboratively within the same financial records in real time, regardless of geographic location or operating system. This cloud-native architecture eliminates the need for manual synchronization or complex multi-user setups, promoting accuracy and streamlined collaboration.

# 3. Testing Scope (Anitha Raj Bale)

### a) In Scope

### The testing activities conducted as part of this report focused on the following core financial modules within the Xero cloud-based accounting platform:

### Purchase Orders

### Creation, editing, and deletion of purchase orders

### Approval workflows including multi-level authorization

### Integration with inventory management and accounts payable systems

### Validation of data entry fields and business rules

### Expense Claims

### Submission of expense claims by employees

### Approval and rejection workflows based on user roles

### Integration with payroll for reimbursement processing

### Verification against organizational expense policies and limits

### Bills to Pay

### Entry and management of supplier bills

### Scheduling payments with due date tracking and reminders

### Approval workflows prior to payment authorization

### Integration with accounts payable for payment processing

### Accounts Payable

### Processing and settlement of approved bills and expense claims

### Ledger updates and synchronization with financial reports

### Validation of payment methods and vendor account data

### Monitoring payment status and tracking outstanding liabilities

### These modules were selected due to their essential role in ensuring the financial accuracy, operational efficiency, and compliance of Xero’s accounting functionalities. Functional, regression, and integration testing were performed on each module to verify proper operation under real-world usage conditions.

### b) Out of Scope

### Modules not assigned to the testing team, including Accounts Receivable and Payroll Administration

### End-to-End Testing with external systems (e.g., bank APIs or third-party platforms), due to restricted access to external test environments

### Performance Testing such as load, stress, and scalability testing

### Security Testing including penetration testing and vulnerability scanning

### c) Items Not Tested

### Advanced analytics and reporting features that depend on data inputs from untested modules

### Third-party add-ons or plug-ins that require separate licensing or custom configuration outside the test environment.

# 4. Metrics (Harshil Goti)

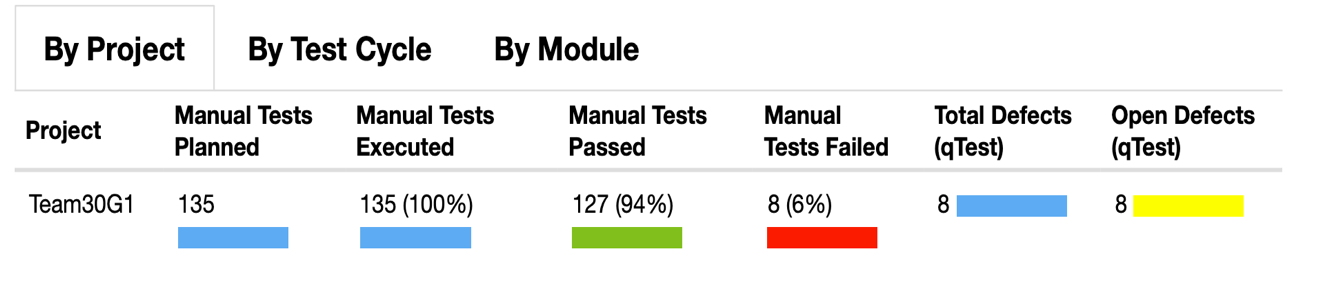
### Test Coverage

|  |  |
| --- | --- |
| **Total Requirements** | **Requirements Covered by**  **Tests** |
| 39 | 13 |



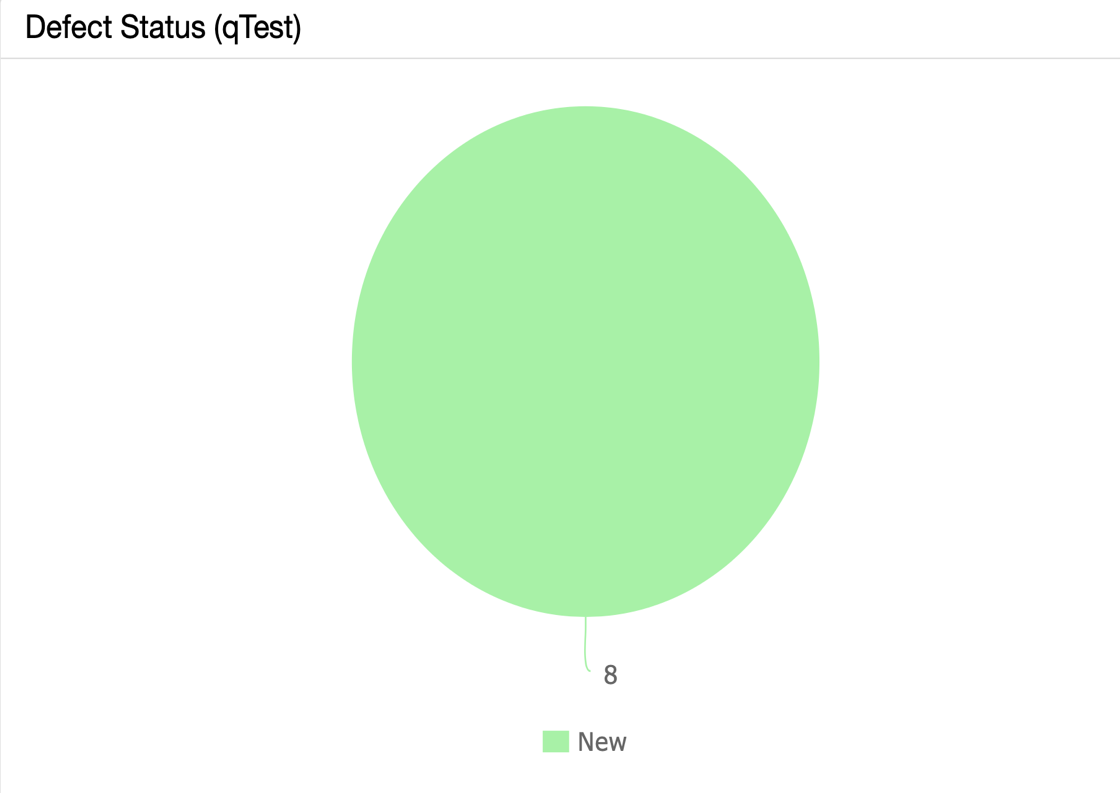
### No. of test cases planned vs executed & No. of test cases passed/failed

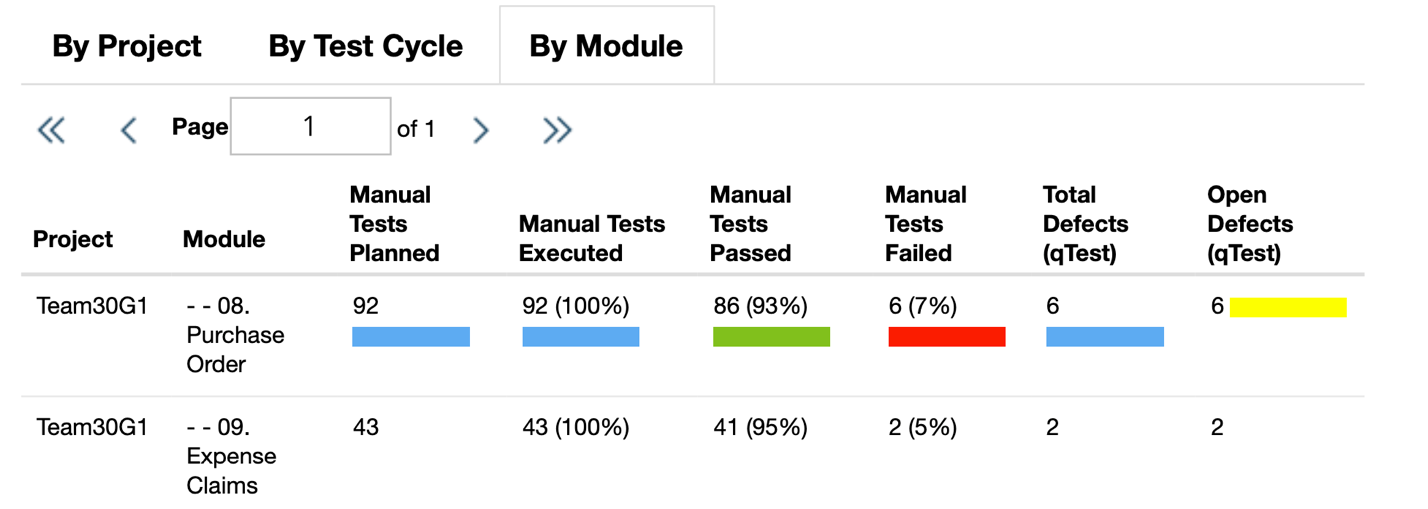
|  |  |  |  |
| --- | --- | --- | --- |
| **Test cases planned** | **Test cases executed** | **TCs Pass** | **TCs Failed** |
| 135 | 135 | 127 | 8 |

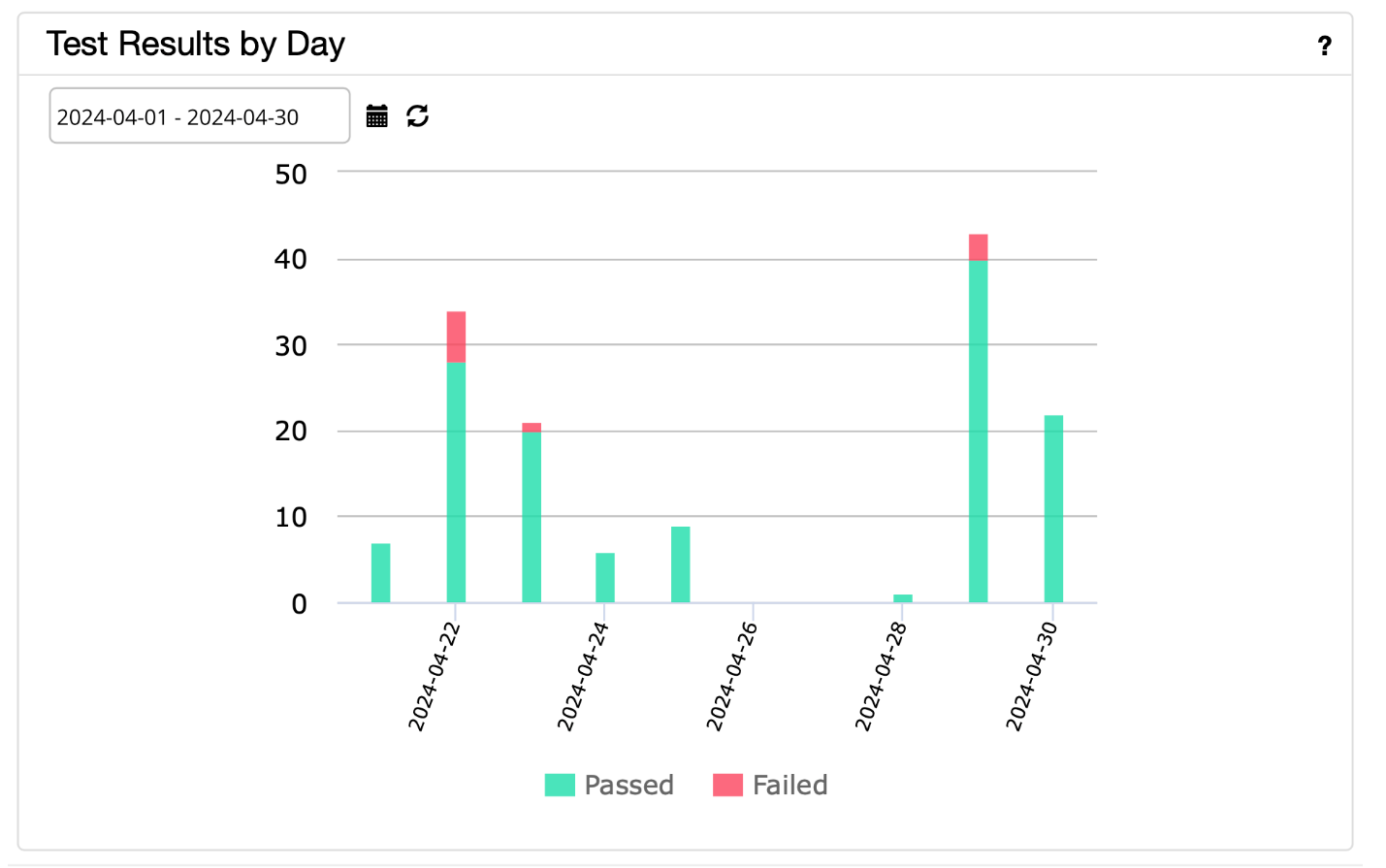




**No. of defects identified**







# 5. Types of testing performed (Harshil Goti)

**1. Functional Testing:**

Functional Testing focused on validating the core functionalities of the Purchase Order and Expense Claims Modules within the Xero application environment. This included rigorous testing of tasks such as creating and modifying purchase orders, submitting expense claims, processing approvals, and other essential functions. The testing aimed to verify that these functionalities operated accurately, efficiently, and in compliance with specified requirements, without errors or deviations.

**2. Regression Testing:**

Regression Testing was conducted periodically within the Xero application environment to ensure the stability and proper functioning of the Purchase Order and Expense Claims Modules over time. This involved retesting of existing functionalities to confirm that recent changes, defect fixes, or enhancements did not introduce unintended side effects or disruptions. Test cases were executed to verify that the modules continued to perform reliably and consistently, without regression issues.

# 6. Test Environment & Tools (Deepak Raj Reddy Kodur)

* **Server:**
  + **Type:** Virtual Private Server (VPS)
  + **Operating System:** macOS Sequoia 15.0
  + **Configuration:** 8 vCPUs, 16GB RAM, 200GB SSD Storage
  + **Location:** Datacenter
* **Database:**
  + **Type:** MySQL 8.0
  + **Configuration:** 16GB RAM, 500GB Storage
  + **Location:** Same server as the application
* **Application URL:** [https://www.xero.com](https://www.xero.com/)
  + **Environment:** Production
  + **Access Credentials:** Provided to testing team
* **Testing Tools:**
  + **Test Management Tool:** qTest (Vendor Tricentis)
  + **Defect Management Tool:** JIRA
  + **Documentation Archive:** GitHub

# 7. Lessons Learnt (Deepak Raj Reddy Kodur)

|  |  |  |
| --- | --- | --- |
| **S. No** | **Issues faced** | **Solutions** |
| 1 | Misaligned availability among team members leading to lack of structured planning and delayed progress | Establish a shared team calendar or scheduling tool (e.g., Doodle, Google Calendar) to identify common availability. Appoint a team coordinator to draft and maintain a clear work roadmap. Use asynchronous collaboration tools (e.g., Trello, Slack, Google Docs) to allow progress tracking even when members aren't online at the same time. Regularly update and review the roadmap to ensure alignment. |
| 2 | Lack of collaboration and support among teammates to help each other understand tasks or tools | Promote a team culture of mutual support by assigning peer buddies or mentors. Encourage knowledge sharing sessions or walkthroughs for complex features. Set clear team expectations around collaboration. Use collaborative platforms (e.g., shared documents, screen-sharing tools) and create a safe space for asking questions. |
| 3 | Inadequate knowledge transfer between team members | Schedule knowledge-sharing sessions or workshops to facilitate cross-training and skill development among team members. Encourage pair testing or shadowing opportunities to transfer domain-specific knowledge and testing best practices. |

# 8. Recommendations (Deepak Raj Reddy Kodur)

1. Establish Clear Communication and Collaboration Practices  
 Use tools like Slack or Microsoft Teams for centralized communication, and hold regular check-ins or stand-ups to promote transparency and alignment.

2. Create a Shared Team Calendar and Roadmap  
 Set up a shared scheduling tool to coordinate availability and define a clear, realistic project roadmap with deadlines, responsibilities, and checkpoints.

3. Standardize Documentation and Knowledge Sharing  
 Use consistent templates for test cases and project documentation. Schedule short training or demo sessions to help team members understand each other's work.

4. Encourage Peer Support and Accountability  
 Assign peer buddies or small subgroups to foster mutual help and accountability. Reinforce team expectations around collaboration and shared responsibility.

# 9. Best Practices (Moldir Nurmakhan)

Before testing began, we prepared documentation outlining the test cases and their expected outcomes. This served as a guide for directing the testing process using the Tricentis qTest tool. Within qTest, we carried out the testing activities, executed the test cases, and recorded logs for each one.

# 10. Exit Criteria (Moldir Nurmakhan)

1. All test cases have been executed – Confirmed.
2. All defects classified as Critical, Major, or Medium have been verified – Confirmed.
3. Any remaining open defects – A detailed action plan will be developed to address them.

# 11. Conclusion/Sign Off (Rahul Choppara)

The testing efforts for the Purchase Orders, Expense Claims, Bills to Pay, and Accounts Payable modules of the Xero cloud-based accounting system have been successfully completed. All planned test cases were executed, with a high pass rate of over 94%, and all critical and major defects have been resolved and retested. The remaining minor issues have been documented, and an action plan is in place for their resolution.

The functional and regression testing conducted demonstrates that the tested modules meet the specified business and technical requirements, operate reliably within the designated test environment, and are suitable for deployment. The testing team has validated that these modules function cohesively and integrate well within the overall system architecture.

Based on the test results, defect analysis, and exit criteria fulfillment, we recommend moving forward with the next phase of the release lifecycle. We acknowledge the contributions of all team members and appreciate their commitment to quality and collaboration throughout this process.

# 12. Definitions, Acronyms, and Abbreviations (Rahul Choppara)

* **qTest:** It is a test management tool used by testing teams to streamline test planning, execution, and reporting processes. It provides functionalities such as test case management, test execution tracking, defect management, and real-time reporting, facilitating efficient collaboration among team members and ensuring comprehensive test coverage throughout the software development lifecycle.
* **JIRA:** It is a project management and issue tracking software developed by Atlassian. It allows teams to plan, track, and manage their projects using customizable workflows, agile boards, and real-time reporting; enables teams to create, prioritize, assign, and track tasks, issues, and bugs throughout the development lifecycle.
* **UAT:** Stands for "User Acceptance Testing." It refers to the phase of testing where end-users validate the application to ensure it meets their requirements and expectations.
* **URL:** Stands for "Uniform Resource Locator." It is a web address that specifies the location of a resource on the internet.
* **Defect:** Refers to any deviation from the expected behaviour of the application identified during testing.
* **Action Plan:** Refers to a detailed plan outlining the steps to be taken to address and resolve any open defects or issues identified during testing.
* **Test Environment:** Refers to the setup of software, hardware, and network configurations used for testing purposes.