Test Summary Report

Team 1

Spring 2025

**Table of Contents**

[1. Purpose (Misha Patel) 3](#_Toc166340969)

[2. Application Overview (Misha Patel) 3](#_Toc166340970)

[3. Testing Scope (Misha Patel) 3](#_Toc166340971)

[4. Metrics (Anvika) 4](#_Toc166340972)

[5. Types of testing performed (anvika) 7](#_Toc166340973)

[6. Test Environment & Tools (Anvika) 8](#_Toc166340974)

[7. Lessons Learnt (Mayank) 9](#_Toc166340975)

[8. Recommendations (Mayank) 9](#_Toc166340976)

[9. Best Practices (Mayank) 10](#_Toc166340977)

[10. Exit Criteria (Shahrukh) 10](#_Toc166340978)

[11. Conclusion/Sign Off (Shahrukh) 10](#_Toc166340979)

[12. Definitions, Acronyms, and Abbreviations (Shahrukh) 10](#_Toc166340980)

# 1. Purpose (Misha Patel)

The purpose of this Test Summary Report is to outline the testing efforts and outcomes related to the Financial Data Entry module of the Xero application. This report focuses on two critical stories: 04. Financial Data Entry, 05. Bank Feeds and Reconciliations, 06. Invoicing, and 07. Accounts Receivable. The summary includes approach refinements, executed test cases, data flow diagrams, RCT coverage, and lessons learned.

# 2. Application Overview (Misha Patel)

# Xero is a New Zealand–based technology company that provides cloud-based accounting software for small businesses. The company has offices in New Zealand, Australia, the United Kingdom and the United States. Xero's products are based on the software-as-a-service model and sold by subscription, based on the type and number of entities managed by the subscriber. The products are used in over 180 countries.

# The Xero accounting software uses a single unified ledger, which allows users to work in the same set of books regardless of location or operating system. It provides automatic bank feeds, invoicing, accounts payable, expense claims, fixed asset depreciation, purchase orders, bank reconciliations, and standard business and management reporting.

# 3. Testing Scope (Misha Patel)

### a) In Scope

1. **Financial Data Entry:**  
   Focused on creating, saving, editing, and updating manual journal entries. Covered currency conversion logic (auto-fetch, manual override), account-type validations, dependency-based field behavior, and connectivity/concurrency handling. Included test stories such as 4.2 Handle Currency Conversions and 4.9 Update Journal Entries.
2. **Bank Feeds and Reconciliations:**  
   Included validation of imported bank transactions, reconciliation rules, chart of accounts updates, and transaction history. Tested functionality to manage bank accounts, view audit trails, and ensure secure connectivity with financial institutions.
3. **Invoicing:**  
   Verified invoice creation, bulk generation, scheduling, and customization. Tested automation features like recurring invoices and reminder emails. Validated multi-currency invoice support, permission-based access to invoice actions, and audit trail consistency.
4. **Accounts Receivable:**  
   Tested full invoice lifecycle: sending invoices, applying payments, tracking status, and issuing customer refunds. Covered features like credit limit checks, reconciliation with bank feeds, and integration with customer records and contacts. Verified recurring invoicing and quote management workflows.

**b) Out of Scope**

* Features related to Xero modules not covered in the current semester (e.g., Payroll, Fixed Assets, Inventory).
* Third-party integrations, mobile app testing, or external tax rate APIs.
* Automation testing; all tests were manual.

### c) Items not tested

* + **Advanced Reporting Features:** Some of the advanced reporting functionalities that require data from modules not covered in this testing phase (like advanced financial analytics) will not be tested.
  + **Third-Party App Integrations:** Specific third-party applications that enhance the functionality of the Purchase Order and Expense Claims modules but require separate setup or subscriptions are not tested at this stage.

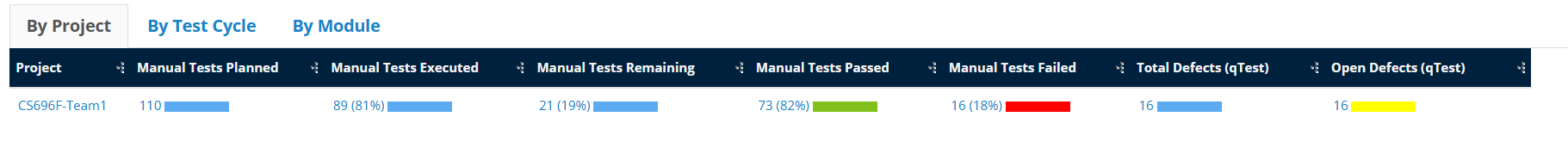
# 4. Metrics (Anvika)

### Test Coverage

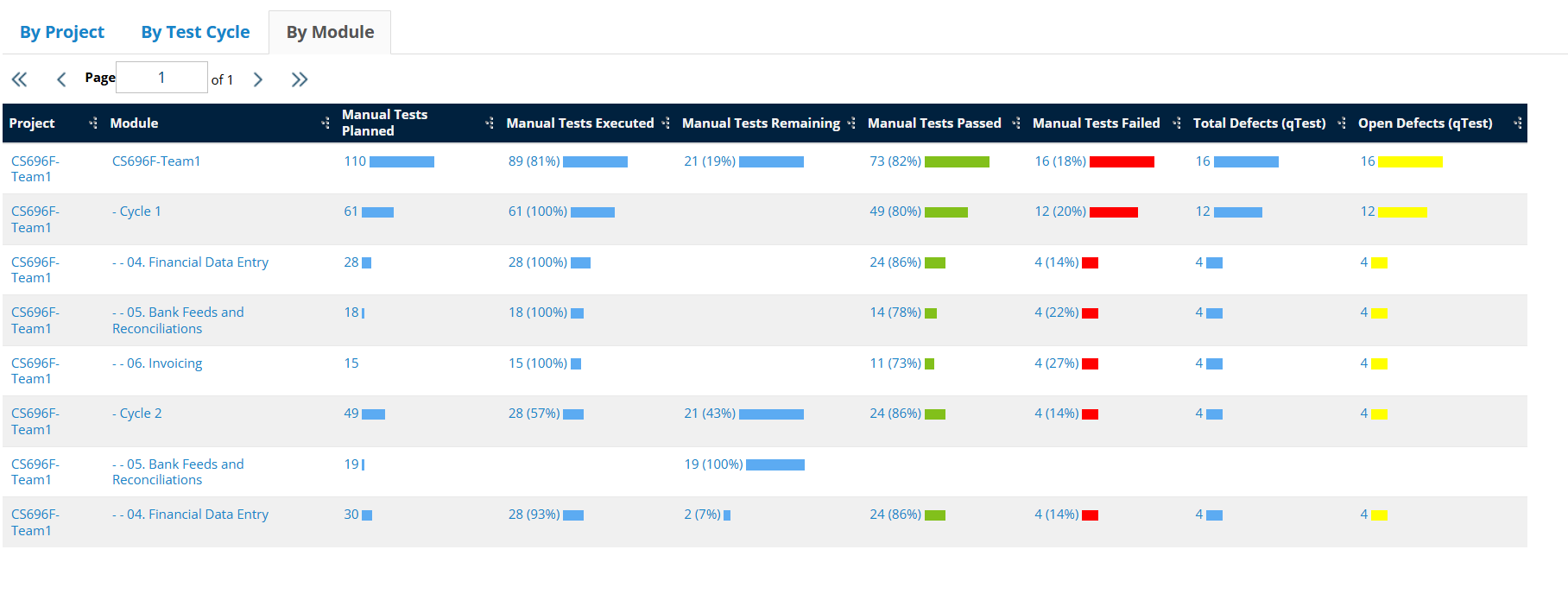
|  |  |
| --- | --- |
| **Total Requirements** | **Requirements Covered by**  **Tests** |
| 55 | 8 |

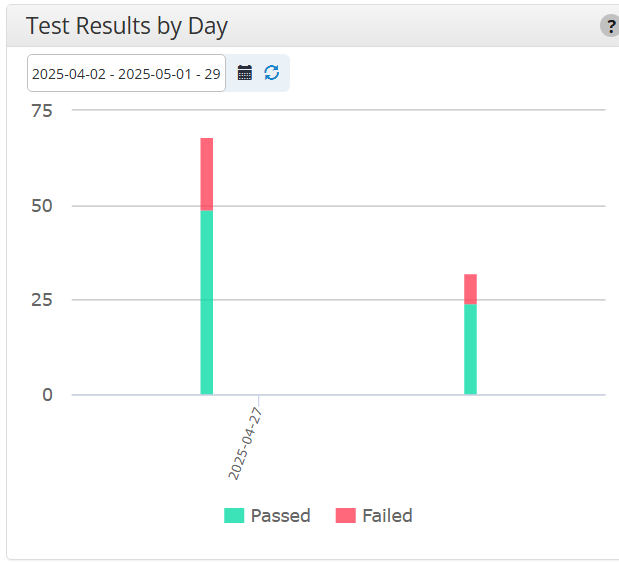
### No. of test cases planned vs executed & No. of test cases passed/failed

|  |  |  |  |
| --- | --- | --- | --- |
| **Test cases planned** | **Test cases executed** | **TCs Pass** | **TCs Failed** |
| 61 | 61 | 49 | 12 |



**No. of defects identified**





# 5. Types of testing performed (Anvika)

**1. Functional Testing:**

Functional Testing was carried out on the Financial Data Entry, Bank Feeds and Reconciliations, and Invoicing modules within the Xero application. The objective was to validate key operations such as entering financial data, importing and matching bank transactions, and generating invoices. Testing ensured that all features worked as expected and met the specified business and system requirements.

**2. Regression Testing:**

Regression Testing was periodically performed to ensure that recent updates or defect fixes did not negatively impact existing functionalities. The team re-executed test cases related to the three core modules to confirm consistent performance, system stability, and functional integrity throughout the project lifecycle.

# 6. Test Environment & Tools (Anvika)

* **Server:**
  + **Type:** Virtual Private Server (VPS)
  + **Operating System:** Linux CentOS 7.5
  + **Configuration:** 8 vCPUs, 16GB RAM, 200GB SSD Storage
  + **Location:** Datacenter XYZ
* **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 (Mayank)

|  |  |  |
| --- | --- | --- |
| **S. No** | **Issues faced** | **Solutions** |
| 1 | Incomplete alignment of test cases with RCT features: Initially, some test cases for *Delete Contacts (7.12)* and *Update Contacts (7.11)* did not fully map to the crosscutting concerns (e.g., Entitlements, Data Validation) defined in the Requirements Coverage Table (RCT). | Conducted a gap analysis to ensure all RCT columns (e.g., ET-Entitlements, FV-Field Validation) were addressed in test cases. Updated test objectives and steps to explicitly reference crosscutting concerns. |
| 2 | Inconsistent test case documentation among team members | Establish standardized templates and guidelines for documenting test cases. Conduct training sessions to ensure all team members understand and adhere to the documentation standards. Regularly review and provide feedback on test case documentation to maintain consistency. |
| 3 | Ambiguity in test case steps for dependency validations (e.g., deleting contacts with linked transactions). | Refined test steps to include explicit validation of error messages and system behavior during dependency checks. Added preconditions (e.g., creating invoices/bills linked to contacts) to ensure realistic scenarios. |

# 8. Recommendations (Mayank)

# Enhance test case traceability: Link each test case directly to its corresponding RCT feature in tools like qTest/JIRA for better visibility.

# Adopt scenario-based testing frameworks for complex workflows (e.g., Delete Contacts with Dependencies) to reduce manual effort.

# Standardize preconditions for test cases involving data dependencies (e.g., pre-creating invoices, recurring payments) to ensure consistency across cycles.

# 9. Best Practices (Mayank)

# Structured Test Design:

# Followed the IEEE 829 standard for test case documentation, ensuring clarity in objectives, steps, and expected results.

# Included Data Flow Diagrams (DFDs) for critical user stories (e.g., Delete Contacts) to visualize system interactions and dependencies during testing.

# RCT-Driven Validation:

# Designed test cases to explicitly validate crosscutting concerns (e.g., ET-Entitlements for role-based access, DDV-Data Dependency Validation for linked transactions).

# Modular Test Data Management:

# Created reusable test data templates (e.g., contacts with/without dependencies) to streamline execution for Update/Delete Contacts and Create Quotes.

# Collaborative Review Process:

# Conducted peer reviews of test cases using qTest to ensure alignment with acceptance criteria and RCT coverage.

# 10. Exit Criteria (Shahrukh)

The following exit criteria were defined and met during this testing phase:

**a)** All planned test cases have been successfully executed.  
**b)** All defects categorized as Critical, Major, and Medium severity have been identified and verified.  
**c)** Any open defects remaining are documented, and a detailed action plan has been prepared to address them post-release or in subsequent testing cycles.

# 11. Conclusion/Sign Off (Shahrukh)

Based on the successful fulfillment of the exit criteria outlined in Section 10, the Testing Team concludes that the application is stable and ready for deployment. The Financial Data Entry, Bank Feeds and Reconciliations, Invoicing, and Accounts Receivable modules within the Xero accounting system have undergone thorough manual testing and validation.

However, it is important to note that final approval should be contingent upon successful User Acceptance Testing (UAT) and Business Acceptance Testing (BAT). These phases are essential to confirm that the application aligns with business needs and stakeholder expectations before full-scale production deployment.

# 12. Definitions, Acronyms, and Abbreviations (Shahrukh)

* qTest: A test management tool used by testing teams to streamline test planning, execution, and reporting. It supports functionalities such as test case management, execution tracking, defect management, and real-time reporting, ensuring comprehensive test coverage and team collaboration.
* JIRA: A project management and issue-tracking software developed by Atlassian. It enables teams to plan, track, and manage work through customizable workflows, agile boards, and detailed reporting. JIRA is widely used for issue tracking and defect management throughout the software development lifecycle.
* UAT (User Acceptance Testing): A phase in the software testing process where end-users validate the application to ensure it meets business requirements and expectations before production release.
* URL (Uniform Resource Locator): The address used to access a resource online.
* Defect: Any deviation from the expected behavior or requirements of the application, identified during testing.
* Action Plan: A detailed set of steps outlining how any identified open defects will be addressed and resolved after the formal testing phase.
* Test Environment: The configured hardware, software, network, and tools setup used to perform the testing activities.