Written Assessment Task

Test Officer, Grade 5/6, Department of Customer Service – Req37374

Daniel Rittie
DANNY@RITTIE.COM / 0478 224 954

Table of Contents

Executive Summary

This document presents a comprehensive plan for addressing an urgent system change for the Long Service Corporation. The testing plan offers a robust, high-quality implementation that minimises risks and aligns with the required organisational standards. The document outlines a detailed approach consisting of preparation, test design, execution, documentation and risk mitigation. Stakeholder engagement strategies are prioritised to collaborate and ensure a smooth communication across several teams, whereas workload prioritisation frameworks such as MoSCoW, are used to focus efforts on high impact areas.

The plan also emphasises measuring success through defined metrics, such as defect rate reduction, timely delivery as well as stakeholder satisfaction. It incorporates post-deployment strategies to assist continuous improvement. Aligned with the organisations commitment to excellence, this approach leverages proven testing methodologies, collaborative strategies and technical expertise to help deliver seamless system updates that can enhance the customer and employee satisfaction.

Objective and Context

Objective

The main objective for this testing plan is to ensure that an urgent system change is properly implemented and thoroughly tested to address any identified issues as well as minimising disruptions to both internal operators and end user experiences. This will involve adhering to established Software Development Life Cycles (SDLC) methodologies, effectively identifying and resolving any defects, and maintaining compliance with the regulatory and organisational standards. The goal of the testing plan is to deliver a seamless, high quality system update that focuses on supporting the Long Service Corporation's mission of enhancing customer and employee satisfaction.

Context

As a newly appointed Test Officer with the Long Service Corporation, I understand the pivotal role this position plays within end-to-end testing of system changes. Given the urgent nature of the change, it is essential to adopt a structured and collaborative approach to ensure all aspects of the system updates are validated and optimised for performance. This will involve collaborating with key stakeholders across technical, operational and customer service domains to identify potential risks, prioritise critical tasks, and keep a clear communication throughout the testing process.

The Long Service Corporation operates at a fast-paced customer-centric environment where system reliability is critical to achieve its broader objectives. The testing plan I produce will be designed to address immediate concerns but also support the Corporation's commitment to innovation, operational efficiencies and a user-friendly experience. By applying my technical skills, understanding of system testing methodologies (i.e. Unit testing, Integration testing, User Acceptance testing etc.) and analytical thinking I aim to contribute a simple to use, resilient system that aligns perfectly with the organisations strategic goals.

Testing Plan – Steps and Recommendations

Firstly, to ensure the successful implementation of the urgent system change, a thorough and structured testing plan is vital. This plan outlines the key steps required, focusing closely on preparation, test design, execution, documentation and risk mitigation. These all aim at minimising defects and ensuring compliance with organisational standards as mentioned in objectives.

Preparation

A good preparation is fundamental to the success of any testing process, the following steps will be taken to gather any necessary information and resources:

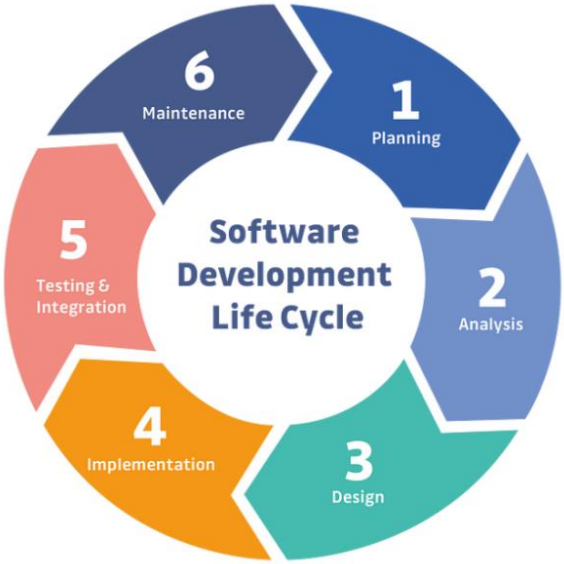
Information Needed:

- Documentation: Obtain detailed functional and technical specifications outlining the system change to fully understand the scope and requirements of the system change.
- Testing Environment Access: A stable and configured testing environment that would mirror the production system as well as the necessary credentials and permissions included.
- Prior Defect Logs and Performance Baselines: Check and review any historical data to identify common issues and establish performance benchmarks.
- Stakeholder Involvement: Collaborate and gain input/feedback from relevant teams to understand their expectations and pain points.
- Business Impact Analysis: Assess any potential implications that the system change may produce to prioritise critical components for testing.

Tools:

- For this role, some crucial tools include Salesforce Test Automation Framework and SAP Test Automation Tools to streamline the testing process. Some other tools that are important can be Jira, to track any defects and platforms such as TestRail for efficient monitoring and reporting.

Preparation forms the foundation of the Software Development Life Cycle (SDLC) which allows for a seamless transition into subsequent phases such as design, testing and deployment. By having documentation, environment setup and stakeholder alignment, the preparation phase directly influences the success of the project. Below is a visual representation of the SDLC and shows how preparation integrates with other stages to achieve a comprehensive and effective development process.



Test Cases

The next step involves making detailed and comprehensive test cases that are tailored to the system change. Key considerations include:

- Functional Testing: To verify the individual components of the system function as expected
- Regression Testing: Ensure the change does not disrupt the already existing functionalities.
- Integration Testing: test the interaction between different modules to confirm a seamless data flow.
- User Acceptance Testing (UAT): Required to validate that the system meets end user expectations and requirements.

It is a priority to focus on high-risk areas such as modules with high user interaction or any critical to business operations. Test cases will be prioritised based on their potential impact on user experience and organisational outcomes.

Executing Tests:

A phased approach will be taken to execute the tests and will ensure validation at every level.

1. Unit Testing: Evaluate individual components or modules for isolated functionality.
2. Integration Testing: test any interconnected components to verify smooth communication.
3. User Acceptance Testing (UAT): Collaborate with stakeholders to check the change aligns with the user needs.

4. Performance Testing: Assess the system scalability and responsiveness under load and traffic.

Jira as mentioned before, can identify any defects in a centralised system and can be categorised by priority and severity. Regular triage meetings must also be held to address critical defects swiftly.

Documentation

Documentation is required to make sure progress is tracked and to support any future development efforts. The test logs will include a format as follows:

Format and Content:

- Test Case ID: A unique identifier for each test case.
- Expected Outcome: The desired result for the test.
- Outcome: The observed outcome during execution of test.
- Defect Severity: A classification of the defects impact (i.e. critical, high, medium or low).
- Resolution Status: Notes on when and how the defect was addressed.

By following this documentation format, it can be kept in a shared repository for easy access and reference for later use.

Risk Mitigation Strategies

To address all potential challenges, the following proactive risk management strategies will be implemented:

Contingency Plans:

- In cases where documentation is not yet available, work with stakeholders and developers to clarify requirements and assumptions.
- Undertake exploratory testing to identify issues that may not be documented.

Escalation process:

- Establish a set and clear escalation hierarchy to ensure that urgent defects are tended to immediately, to minimise delays in the testing timeline.

By following a structured testing plan such as the one above, I aim to deliver a robust and reliable system update that will meet expectations while also minimising any disruptions/risks. This approach ensures thorough testing of the urgent system change, therefore aligning with the Department of Customer Services commitment to excellence.

Stakeholder Engagement Plan

An effective stakeholder engagement is crucial for the success of the testing process, especially in this situation of an urgent system change. By engaging the right stakeholders it ensures alignment, keeps smooth communication and promotes collaboration across multiple levels of the organisation. This plan displays the strategies for identifying and engaging stakeholders to ensure successful implementation and effective testing in 5 main areas.

1. Identifying Stakeholders

Firstly, to conduct any comprehensive testing and implementation, it is necessary to identify all relevant stakeholders who will be impacted by or can influence the system change.

Internal Team:

- Developers: Developers are responsible for making code changes and fixes based on the feedback from testing.
- Project Managers: Project managers oversee the project timelines, resources and deliverables.
- Business Analysts: Business Analysts provide insights into the business requirements and help interpret technical specifications.

Broader Organisation:

- Customer Service: The frontline staff who interact directly with customers and give valuable feedback relating to system issues and usability.
- Leadership: Includes any department heads/executives who need to be informed of the progress and any potential risks.
- End Users: Both internal users and external customers who will directly use the system.

2. Engaging with Team Members

Engagement with the direct team is crucial for a synchronised effort and efficient problem solving for the system. Ways to do this are as follows:

Weekly Stand-Ups for Alignment:

- Purpose: To give regular updates, discuss any challenges and align on priorities.
- Approach: Schedule brief daily/weekly meetings depending on urgency. Encourage open communication to quickly identify and solve any issues.

Collaborative Tools

- Defect Tracking with Jira:
  By utilising Jira to log any defects, assign tasks and track progress in real-time. All team members must have access and be trained to use the tool effectively.
- Documentation with Confluence:
  Maintain updated documentation, this includes test plans, test cases and any meeting notes. Confluence can act as a centralised knowledge base accessible to the team.

Communication Channels:

- Establish group chats/channels through platforms such as Microsoft Teams or Slack for instant and clear communication. Use emails for formal communication and summarising meeting outcomes.

3. Engaging Broader Stakeholders

Broader Stakeholder engagement ensures that the system change meets the organisational needs and if any concerns, then they can be promptly addressed. Some effective ways to capture this is through regular updates whether via email or dashboards. Weekly emails can provide progress reports and highlight any key achievements or upcoming tasks as well as tailor the communication style to the recipient for further clarity. Virtual dashboards using tools like Power BI or Tableau, can be used to display testing metrics, trends or project status.

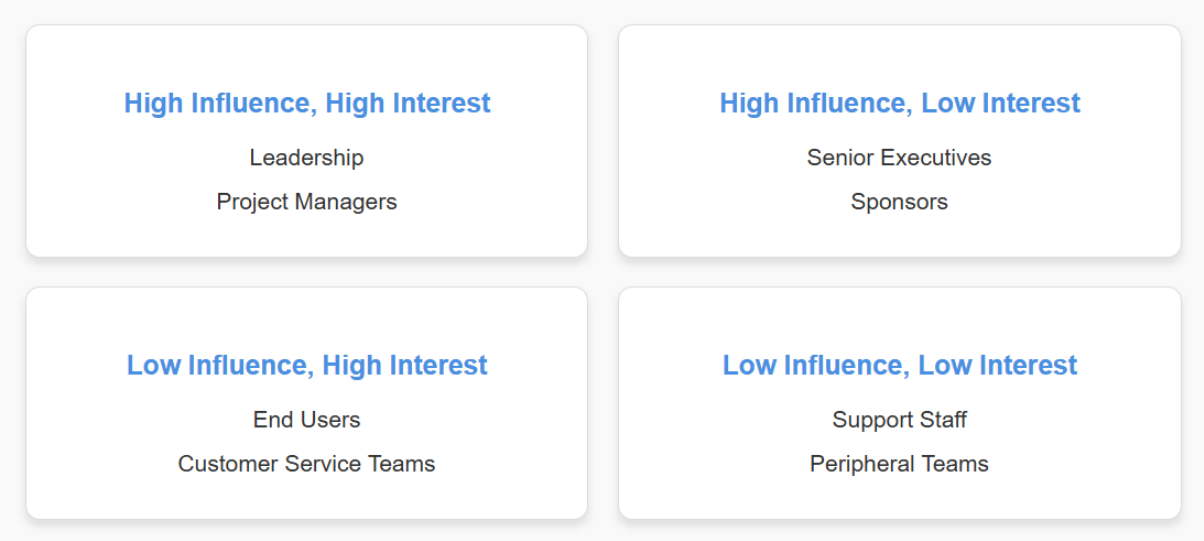
Another common way to engage the broader stakeholders can be through surveys or meetings. Short surveys to gather information regarding system performance/user satisfaction can capture important immediate feedback. Meetings with stakeholders also help obtain approvals when necessary and engage with cross-functional teams to make sure all perspectives are considered.

4. Customer Feedback

By incorporating customer feedback, it not only ensures that the system meets user expectations but also satisfaction. There are many ways to go about gathering customer insights for example, a Post-Implementation Survey to assess their experience after the system change and can include questions regarding system usability, performance or any issues they encountered in their experience. Customer Support Channels analyse trends to identify common issues and monitor support tickets related to the system change to fix any issues promptly. Feedback Integration is also an effective tool to incorporate a chance for the customers/users to provide feedback during the testing process to make sure that user suggestions can be evaluated and when feasible, implemented.

5. Visual Representation

To properly communicate the stakeholder engagement process, a stakeholder map will be used to illustrate relationships and communication flows. Below is a stakeholder map that categorises and visually displays the stakeholders based on their influence and interest in the project:



Stakeholder Map:

- Purpose: To visualise the key stakeholders, their interests and how they interact with each other in the testing process.
- Components: The components include Nodes which represent stakeholders (i.e. customer service, end users and developers), and Connections, which indicate the communication channels and number of interactions.
- Benefits: The benefits consist of identifying potential communication gaps and it facilitates a better understanding of stakeholder roles/influence.

By proactively engaging the stakeholders through structured communication and collaboration strategies, we can ensure that the urgent system change testing is efficient and aligns with the needs of all the parties involved. By doing this approach, it not only mitigates risks but also enhances the overall quality and acceptance of the system, thus contributing to the Long Service Corporations commitment to excellence as well as customer satisfaction.

Prioritisation and First Steps

1. Immediate Actions

To address the urgent system change effectively, the following immediate steps must be taken:

- Analyse Change Documentation: Review any technical and functional specifications to understand the scope, urgency and potential impacts the changes made or will make.
- Set up Testing Environment: Configure and validate the testing environment to make sure it mirrors the production settings. This may include loading necessary data and verifying access to testing tools.
- Kickoff Meeting: Hold a meeting with key stakeholders, including developers, business analysts and project managers. This will help align objectives, priorities and timelines.

2. Workload Prioritisation

To best manage workload efficiently, I will apply the MoSCoW prioritisation framework:

- Must-have: Critical functionalities that are essential to the system operations and user satisfaction.
- Should-have: Features that enhance user experience but would not be classed as immediately critical.
- Could-have: Minor changes or low-risk areas.
- Won't-have: Non-essential items deferred for future consideration or projects.

By doing this, the priority will be given to resolving high-severity defects and testing high-risk areas that will significantly impact business operations. Below is an example Workload Prioritisation Table, further illustrating the application of the MoSCoW framework in this testing plan.

Priority Level	Description	Examples
Must-have	Critical functionalities essential for system operations and user satisfaction.	• Login functionality • Database connection stability • Real-time transaction processing
Should-have	Features enhancing user experience but not immediately critical.	• Improved search filters • Customisable user interface themes • Integration with third-party tools
Could-have	Minor changes or low-risk areas.	• UI color scheme adjustments • Adding tooltips to less-used features • Background animation effects
Won't-have	Non-essential items deferred for future consideration.	• Social media sharing functionality • Advanced analytics dashboard • Integration with future APIs

3. Resource Allocation

Tasks will be assigned based on team members expertise and tool availability:

- Developers will focus on defect resolution.
- Testers will execute prioritised test cases.
- Business Analysts will validate that changes align with business requirements.

By having a structured approach such as this, it ensures an efficient and focused response to the urgent changes.

Measuring Success

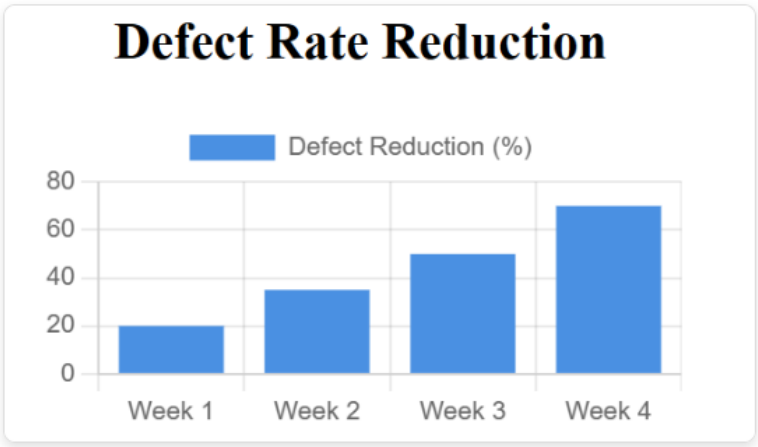
To effectively measure success, it is important to hold it in 2 main categories, 'Key Success Metrics' and 'Continuous Improvement'. Key success metrics, evaluate the effectiveness of the testing process, the following metrics are monitored:

- Defect Rate Reduction: Fewer then two high severity defects post deployment.
- Timely Delivery: Adheres to the set project timelines, especially regarding the critical testing phase.
- Stakeholder and Customer Satisfaction: Positive feedback through post-deployment stakeholder meetings and surveys.
- Standards Compliance: Full compliance with all internal policies and industry standards.

Continuous Improvement focuses on the post-deployment side and is conducted to identify opportunities for process optimisation:

- Analyse defect trends to uncover any recurring issues.
- Update testing frameworks and strategies based on lessons learned.
- Document insights to help guide future urgent system changes.

This will ensure ongoing refinement of the testing process and supports an uplifting culture of continuous improvement.



Example KPI Dashboard: Defect Rate Reduction over four weeks.

Alignment with Capabilities

Resilience and Courage

Handling urgent changes requires a large amount of composure and adaptability. During my academic projects, I navigated tight deadlines while implementing complex solutions, such as troubleshooting API integrations while under time constraints, and helping a Photography Business with a Python script to massively cut down back-end process time. These experiences prepared me to remain calm and solution focused under any pressure.

Commit to Customer Service

By centering testing efforts on user experience, I ensure systems meet customer needs. For example, during my fantasy football website project, I incorporated user feedback to enhance the interface and provide the user a seamless experience. While working for my previous employer at LookPro Photography, I also assisted customers with any problems they faced while trying to navigate the website or with orders.

Work Collaboratively

I am a big believer that collaboration is key to effective testing. I have worked in cross-functional teams during university group projects, utilising tools such as Jira to track tasks or Microsoft Teams to work collaboratively. These practices ensured alignment and a focus on progress.

Think and Solve Problems

My ability to analyse and resolve issues is demonstrated by automating repetitive tasks at LookPro Photography. By utilising and making Python scripts, I reduced process times significantly, perfectly showcasing my problem-solving skills.

Technology

Due to my history and comfort in an array on programming languages and frameworks, I am equipped with the technical expertise and fast learning needed to adapt to new technologies. I have a wide range of knowledge using tools for performance and testing such as JMeter and Wireshark. By having hands-on experience in these tools and areas, it assists with the thorough executions required for testing.

Conclusion

In conclusion, this plan demonstrates a structured, professional approach to managing and executing the testing process for an urgent system change. By focusing on preparation, stakeholder engagement, prioritisation and risk mitigation, the plan ensures minimal disruption with maximum efficiency.

By utilising measurable success metrics and a commitment to continuous improvement, I aim to deliver a reliable, user-centric system update that supports the Long Service Corporation's broader objectives of innovation and operational excellence. My experience, technical proficiency and collaborative mindset position me to strive and contribute to the organisations mission of enhancing customer and employee satisfaction.

I am eager to bring my skills and dedication to this role and look forward for the opportunity to contribute to the success of the Long Service Corporation.