

Quality Management Plan (Sample)

Optum Rx will provide a quality management plan to support the Configuration, Implementation, Deployment, and Operation of the systems outlined. Optum Rx's Quality Control process aims at continual improvement of the solution and to uphold quality standards. The Plan-Do-Check-Act process that will be followed for all quality management tasks is outlined below:

Plan: By establishing the appropriate objectives and processes to deliver the required results, we meet output expectations with accuracy and thoroughness.

Do: This step in the process assures the process executes properly and the product meets the expectations.

Check: An integral part in the four-step method is studying the actual results. The actual results are compared against the expected results to discover any differences. During this step, the system is checked for appropriateness and completeness.

Act: Corrective actions are performed on significant differences between the actual and planned results. Differences are thoroughly examined to determine the root cause and necessary changes are applied to improve the solution.

As part of the Quality Management program, we will also produce the master test plan describing the quality goals, test strategy, types of testing performed, test automation strategy, tools used, resources required, entry and exit criteria and a process for the State's review and approval.

We maintain an independent Quality Management organization staffed by dedicated personnel with specific training and expertise in failure mode analysis, root cause analysis, error prevention, continuous process improvement, quality assurance assessments and quality process automation. Quality Management support spans Optum Rx pharmacy care and pharmacy benefit management services. The department conducts over 35,000 monthly process and product audits for multiple functional areas across operations. Audits assess process and product adherence to regulatory requirements (federal and state), contractual requirements (client and provider) and internal process requirements. Audits are independent, conducted within a standardized internal audit framework and regularly calibrated. Audit results are consistently monitored, analyzed, and reported out to the business.

Organizational Strategy

Optum Rx Quality Management exists to drive operating performance—one of our company's five core priorities. We achieve this through investing in the training and development of our talent, the delivery of independent internal audits and the development of data driven quality improvement solutions. Quality Management operates under the idea that you can only improve what you measure, and we focus all our efforts on measuring what matters, our organization's ability to meet and exceed customer and client expectations.

The key to our success is the continuous focus on improving our service delivery model to increase flexibility, reliability, and scalability. Achieving a right the first-time culture to

deliver on our commitments to our clients requires the engagement, collaboration and leadership of everyone at Optum Rx. Throughout our enterprise, our processes, standards and behaviors reflect this culture.

Our quality assurance program for support of defects and enhancements is embedded directly in the software development life cycle deployed at Optum Rx. We follow a process that features formalized requirement gathering, along with SME and client signoff on all design and specification work. Afterwards, we conduct an auditable and documented testing process that includes the following:

- Unit testing (including peer review)
- Quality assurance testing (assisted by automated test tools)
- Requestor acceptance testing
- Integration testing and installation testing
- Parallel regression testing (assisted by our proprietary RxPARALLEL tool)
- A pre-production beta test program.

We designed these steps to validate that our software releases are thoroughly tested for defects and applicability to stated requirements and specifications. This testing process also makes certain that:

Users have the ability to see and experience changes before they are introduced into production.

- The software meets performance-related requirements.
- The installation procedure is thoroughly tested before being used in production environment, validating the quality and timeliness of the installation process.

Test Plan Development

We compose and submit test plans for each software modification that we undertake. Proving and documenting the applicability of the modified code and its continued applicability for non-affected functions is paramount in the process.

The test plan specifies that enhancements and changes to our system are exposed to three different sets of tests that validate the functionality of the desired change and the proper operation of the system. Three specific testing scenarios are used:

- Unit testing of the software for conformance to meeting stated requirements
- QA testing of the change to demonstrate functionality and compliance with systems standards
- Requestor testing exposing new functionality to the user or user agent to validate that requirements and usability expectations are met.

In addition to the test cases that include boundary conditions that are part of the design process for a modification, we also employ an automated test tool that allows for standardized testing procedures and leveraging archived test scenarios.

Regression and Scalability Testing

We also expose all changes, after integration testing, to an aggressive parallel and regression testing using our comprehensive test tool. Finally, we expose all code to our beta test environment before it is approved for true production deployment.

As part of the software development life cycle (SDLC) for our software products, we perform an extensive set of regression test procedures that compares the results and the performance of each new software version against known benchmarks.

For accuracy and compatibility, we have designed a set of comparisons that the software is subjected to while we incorporate test scenarios from our production environments to confirm that any variation in results can be inspected and verified as being expected and desirable.

For performance testing, we compare each new version against known production versions with a number of performance measurements. The elapsed time for operations, the amount of system resources, the number of I/O operations and other metrics are compared when the versions are exposed to the exact same workload. This provides an excellent indicator for the quality of a software release and allows us to plan for any expected variations.

Manual and Automated Testing Capabilities

Enhancements and changes to our system are exposed to three different sets of tests that validate the functionality of the desired change and its proper operation of the system. The three specific testing scenarios are used:

- Unit testing of the software for conformance to meeting stated requirements
- QA testing of the change to demonstrate functionality and compliance with systems standards
- Requestor testing exposing new functionality to the user or user agent to validate that requirements are met and the functionality meets usability expectations.

All three of the testing scenarios use a third party software test tool that allows for collection and documentation of test cases. It also provides the ability to drive the examination of the test cases and evaluation of results. In addition, the tool serves as a documentation repository for testing documentation, test cases, test results, and the approval of key constituents of the test results.

In addition to the testing described above, we provide a robust regression testing approach, process, and environment through the use of our own proprietary tool which allows us to test new versions of our software against existing production versions of the same application. These parallel tests allow us to expose millions of near-production transactions against a baseline database using the “known” software version as well as the new software upgrade. The execution of these tests and evaluation of the results is automated and details any discrepancies.

Examples of testing results include:

- Claim status results

- All (example, accepted and rejected) transaction response data (including reject codes and messaging) reported back to the pharmacy
- All financial details of the transaction including pricing, copayment calculations, accumulations, COB results.
- All data to be captured in the data warehouse
- In addition to the consistency of the data results from adjudication, our proprietary tool also allows for the evaluation of the performance of the new software. This evaluation allows us to examine:
 - Comparisons of transaction times
 - Comparisons of elapsed times for entire test cycles
 - Comparisons of system resource usage (example, CPU utilization, I/O requirements, memory utilization)

With the information provided, we accurately predict reliability, performance and ultimate success of our newly developed software.

We perform testing within its own environment while using actual claim, benefit, and member data. This service differentiates Optum Rx in the marketplace by providing our clients with the assurance that software upgrades to the claim processing system will have the desired outcomes. As a result, changes to the claim processing system are seamless to our clients and their members.

Quality Metrics

Using our quality testing capabilities and our requestor acceptance and beta test programs makes certain that quality software and installation processes are deployed.

After going through our rigorous development life cycle, the defect rate experienced for new software is less than 2 percent. The defect rate experienced for those items due to an error in requirements is less than 1 percent.

Data Accuracy

Our average error rate on system development/coding changes is impressively low. As a result of our formal design review and development process, our extensive multi-pronged approach to testing (including formal regression and automated parallel tests), and the structured beta test period, the number of defects found in code that has been made available for general release is extremely small. The percentage of change requests that are deemed defective is consistently below 2 percent, measured across all changes and across all reported environments. An individual customer typically experiences an error rate far below even these levels.

Test Strategy and Approach

Our goal is to be our clients' trusted champion in achieving wellness, and to create a "right the first time" client-focused culture in everything we do. To help achieve this, our Agile delivery teams manage our quality assurance and continuous process

improvement programs. Our support of defects and enhancements is embedded directly in the software development life cycle deployed at Optum Rx.

Achieving a “right the first time” culture to deliver on our responsibilities to our client requires the engagement, collaboration and leadership of everyone at Optum Rx. To achieve and maintain this engagement across the entire enterprise, we provide training opportunities for individuals at all levels of the organization, including front line employees and individuals in leadership roles.

Quality is the entire team’s responsibility. Our quality assurance program for support of defects and enhancements is embedded directly in the software development life cycle deployed at Optum Rx. We follow our OSAM (Optum Scalable Agile Methodology). It is a scalable, lean, metrics-driven approach to software delivery that fosters engineering and quality excellence and agile maturity at any scale. By simplifying our processes and providing a blueprint for highest quality delivery using OSAM we are able to reduce our delivery cycle time, raise the quality of our solutions, improve the predictability of our releases and increase the productivity. We are also able to provide transparency into our project health.

The Optum Rx test strategy is used by our core Agile teams across the organization that manage testing. Our strategy and approach describe how and when the testing is conducted, who will do the testing, the type of testing being conducted and what tools are used to support automation and defect management.

For each of our products, there is a test strategy prepared by the agile core team:

- Feature testing occurs throughout each sprint with the final user acceptance testing (UAT) conducted in the testing sprint before going live.
- During the release planning meeting, the team will capture acceptance criteria and immediately add them as logical test cases linked to the product backlog item.
- In each sprint planning meeting, a sprint test plan will be created and reviewed.
- As the testing is done in each sprint, the results will be tracked along with which features have been successfully tested and which ones have defects.
- Special attention is given to distinguish defects from changes in requirements. If the team requires a change in requirement, it will be added to the product backlog and prioritized.

Types of Testing

Different types of testing occur at different points within the software development lifecycle. Each type of testing serves a different purpose, but all build up to final user acceptance testing before the system goes live.

Unit Testing

Unit tests are conducted to confirm an operational baseline install of a software component or to confirm connectivity to another system. Unit tests are required whenever a software component is installed into an environment. While testing of the

installation in the development environment would be adequate to validate the installation procedure prescribed for the solution, any installation into any subsequent environment must be appropriately validated. This testing is conducted by the agile core team members, along with the help of selected technical support staff.

State of Georgia

Quality Management Plan

Prepared by: Optum Rx
V 1.0 | 03/01/2024

Revision log

Version	Primary author(s)	Description of version	Date completed
<Rev XX>	<Author 1, Author2>	<Description>	<MM/DD/YYYY>
<Rev XX>	<Author 1>	<Description>	<MM/DD/YYYY>

Contents

Revision log	3
Contents	4
Figures	5
Tables	6
1 Introduction	7
1.1 Objectives	7
2 Roles and responsibilities.....	8
3 Quality management	9
3.1 Software configuration management	9
3.2 Work product audits.....	9
3.3 Quality management audits (reviews and assessments)	10
3.4 Tracking metrics	12
3.5 Monitoring requirements traceability.....	12
3.6 Testing standards and practices	13
3.6.1 Unit testing.....	13
3.6.2 Integration testing	13
3.6.3 System testing.....	13
3.6.4 User acceptance testing.....	14
3.6.5 Stress testing.....	14
3.6.6 Parallel testing	14
3.7 Phase closeout project reviews	14
3.8 Continuing support.....	14
4 Tools, techniques and methods	15
4.1 Deliverable quality assurance.....	15
4.2 Document repository	15

Figures

No table of figures entries found.

Tables

<i>Table 1: POS Quality Management Team Roles and Responsibilities.....</i>	<i>8</i>
<i>Table 2: Sample of Auditable Deliverables/Artifacts</i>	<i>10</i>

1 Introduction

The Optum Rx Pharmacy Point-of-Sale (POS) Team has developed a comprehensive and thorough Quality Assurance (QA) methodology that follows a set of defined and repeatable processes, enabling us to monitor the project in a manner that promotes continuous quality improvement.

The POS Team members, in conjunction with functional and technical staff, have collaborated to develop quality management strategies that provide oversight and monitoring during all phases of the project and ensure all contractual obligations are being met. Our goal is to regularly measure our performance, identify opportunities for refinement based on the analysis of established metrics, and then implement changes in our activities designed to achieve measurable improvements.

1.1 Objectives

The Quality Management Plan supports the Design, Development, Implementation (DDI) project phase and the operation of the systems outlined in the State of Georgia RFP.

This plan is designed to guide quality-related activities during application development and deployment, and to ensure that the implementation of the pharmacy POS system follows predefined, repeatable processes.

The POS Team's QA methodology is designed to meet the following goals:

- Produce quality deliverables and documentation
- Improve software quality by monitoring both the software and the development process that generates it
- Ensure full compliance with the established standards and procedures for the software, the software processes, and ongoing support activities
- Ensure that any inadequacies in the process or product, as it relates to standards, are brought to the attention of the POS Executive Management Team and addressed expeditiously

2 Roles and responsibilities

Independence and reporting relationships are key components in producing a successful Quality Assurance Plan. Team members should fully understand their role and responsibilities within the project, and by virtue of this awareness, they will be equipped to execute their duties with confidence.

Table 1 describes the POS Quality Management Team's roles and responsibilities.

Table 1: POS quality management team roles and responsibilities

Role	Responsibility
POS quality assurance manager	• Develops quality assurance policies and procedures to ensure high quality outcomes are achieved • Monitors system results and ensures they meet the expected results • Evaluates and verifies ongoing activities for performance, effectiveness, and accuracy • Responds to quality assurance issues
POS account manager	• Reviews all major deliverables to ensure quality standards and Department expectations are met
POS project manager for the project management office (PMO)	• Reviews all deliverables to ensure quality standards and Department expectations are met • Coordinates with the POS Process Engineer and the POS Technical Leads to schedule reviews and audits • Identifies artifacts for review and audit • Identifies review participants • Incorporates reviews, audits, evaluations, and test activities into the project schedule • Tracks risks, issues, and defects that emerge from these activities and ensures they are resolved
POS technical leads	• Documents project-specific guidelines relevant to their disciplines • Identifies peer reviewers based on team member skill set • Assists in identifying key artifacts for inclusion in review and artifact sample groups • Performs audits of artifacts and processes • Identifies and documents non-compliance and opportunities for improvement
POS technical peer reviewers	• Reviews artifact content to identify errors or problems

3 Quality management

The POS Team's approach to quality is defined by the following:

- Software configuration management
- Work product audits
- Quality management audits (reviews and assessments)
- Tracking metrics
- Monitoring requirements traceability
- Testing
- Phase closeout project reviews

3.1 Software configuration management

The Optum Rx POS Team has robust testing and code promotion procedures in place that adhere to the software development life cycle (SDLC) best practices from organizations like the Institute of Electrical and Electronic Engineers, Inc. (IEEE®) and the Project Management Institute, Inc. (PMI®). The SDLC workflow is supported by the use of issue and project tracking software, source control, and a central automated build server.

We maintain separate development, testing, and production environments with established controls in place to ensure code promotion and system deployments are thoroughly documented and deployment procedures are tested before being promoted to production. The POS Team ensures that system changes are reviewed and approved prior to being transferred to a deployment team who follow the deployment documentation in order to promote new code and data into production.

The POS Team's Development Policy includes strict version control procedures that are applied to all software artifacts, including applications, back-end systems, and modules, with each clearly defined in their specifications and consistently versioned throughout. Features are tracked by product version, including all supporting use cases and test cases that are cross-referenced back to the original requirements.

Software artifacts are configured to be built using a system that identifies the specific artifact's version as well as any dependencies. The actual software builds and packaging are performed by the POS Team's automated build server to ensure that all builds contain only software that is under version control, and that all builds are fully automated and reproducible. A comprehensive suite of unit tests that tie directly back to use cases, and cross-reference to the original requirements, are executed before and after each update to identify differences. These tests must fully pass before a build can be published, and further, full regression testing is performed before a code change or new software version is deployed to production.

3.2 Work product audits

Audits are independent examinations of deliverables that are performed on a standard set of POS artifacts, including those that have been identified within the RFP, for the purpose of verifying that project results have met the expected standards and outcomes.

To ensure that the most complete version of an artifact is being audited, while still allowing for any identified non-compliance issues to be addressed, audits are performed at the same time as the final peer reviews prior to artifact approval and the creation of the quality assurance baseline.

3.3 Quality management audits (reviews and assessments)

Table 2 identifies a sample of artifacts and deliverables that will be subject to internal quality assurance audits; however, this is not an all-inclusive list. For each deliverable or artifact to be audited, the following will be identified and documented when applicable:

- Full name of deliverable or artifact
- POS Project Manager for the PMO
- Project Director
- Designated deliverable author
- Designated document reviewer
- Date of request
- Comments/special requests

Table 2: Sample of auditable deliverables/artifacts

Artifact/deliverable	Audit (Y/N)	Date
Start-up tasks		
Communications Management Plan		
Project Work Plan		
Risk Management Plan Implementation		
Integration Management Plan		
Quality Management Plan		
Staffing Management Plan		
Applications Implementation Plan		
Interface Plan		
Analysis and design Tasks		
Architectural and System Requirements Document		
Joint Application Design (JAD) Session Results		
POS System Development Methodology		
Requirements Specification Document		
Requirements Traceability Matrix		
Detailed System Design		
Revised Interface Plan		
Test Management Plan		
Disaster Recovery Plan		
Configuration Management Plan		

Artifact/deliverable	Audit (Y/N)	Date
Development tasks		
Updated Configuration Management Plan		
Business Rules Documentation		
Business Rules Use Cases		
Business Rules Output Reports		
Updated Interface Plan		
POS System Documentation		
POS User Documentation Manuals		
Testing tasks		
POS System Test Schedule		
POS System Test Plans		
UAT Test Plan		
Operational Readiness Test Plan		
POS System Test Cases and Scripts		
UAT Test Cases and Scripts		
ORT Test Cases and Scripts		
System Test Results		
Unit Test Results		
Rules Test Results		
Module Test Results		
UAT Test Results		
ORT Test Results		
Test Tracking Reports		
Performance Tuning Document		
UAT Readiness Certification		
Final Testing Report		
Operational Readiness Report		
Updated Requirements Traceability Matrix		
Updated Detail Design Document		
Training tasks		
Knowledge Transfer Schedule and Plan		
Knowledge Transfer Documentation		
Knowledge Transfer and Application Software		
Knowledge Transfer Operational Certification		
Final Knowledge Transfer Schedule and Activities		
Knowledge Transfer Completion Certification Letter		
Knowledge Transfer Reports		

Artifact/deliverable	Audit (Y/N)	Date
Implementation tasks		
Implementation Report		
Updated Contingency Plan		
Production Schedule		
Report Distribution Schedule		
Backup and Recovery Procedures		
Security Manual		
Final Implementation Checklist		
Final Implementation Report		
Implementation Certification Letter (Ready for Operations)		
Transition to Operations Plan		
Post Implementation tasks		
Post Implementation Evaluation Report		
All Required CAPs		
CMS Certification Documentation (First Run)		
Post Implementation System Documentation		
Post Implementation User Documentation		
Updated CMS Certification Checklists		
CMS Certification Readiness Checklists		
Required information and Documentation		
Revised System Documentation Resulting from System Remediation		

3.4 Tracking metrics

The Optum Rx POS Team will track the following metrics:

- Total requirements validated by date span (Business Rule closure status)
- On-time delivery to client of project deliverables
- Number of open/closed deliverables by project phase and whether they are behind schedule
- Status of action items by person and by status

3.5 Monitoring requirements traceability

The POS Team will create a crosswalk of RFP requirements to Business Rules (BR) through detailed review of the RFP, contract, related documents, and the use of our in-house created requirements management tool to track functional requirements through the life cycle of the project.

Within this process of monitoring and managing requirements, the BRs represent clarifications and decisions, based upon the RFP requirements, that are documented, managed, and are critical to project success. Once BRs are approved,

they are delivered to the POS Technical and Development Teams to guide the design, development, configuration, and deployment of systems and services, minimizing rework and requiring all issues to be resolved and documented.

By tracing requirements back to the RFP, the POS Team and the Department can be assured that each has been addressed and that there are no gaps in functions or services. The POS Team has the ability to create new requirements as a result of contract discussions, amendments, or approved JAD changes through the formal Change Management Process.

3.6 Testing standards and practices

Consistent with the Agile software development model, our software development and deployment process leverages test-driven development (TDD) and continuous integration. Software is developed based on specific test cases which can be explicitly traced back to requirements. Projects are set up to utilize source control with an automated build system and a full suite of unit tests, which are invoked prior to each build.

During the DDI phase of the Project, the POS Team will perform several levels of internal testing and submit the draft Test Plan for approval by the Department. The plan will address unit testing, pharmacy POS system and integration testing, and user acceptance testing, and include a template cross-reference for the requirements traceability matrix.

3.6.1 Unit testing

As part of TDD, all POS software includes a full set of unit tests. As developers implement modules, they also write unit tests in order to validate that these modules fully satisfy all of their corresponding test cases. The suite of tests typically include “negative” tests in order to ensure that each module behaves appropriately even with invalid or unexpected inputs. The tools employed during unit testing includes:

- JUnit
- HtmlUnit
- HttpUnit
- Selenium IDE

All unit tests are executed and verified with every code change, and only when an application has passed 100% of its unit tests can it be promoted to the next environment to perform integration testing.

3.6.2 Integration testing

Integration testing is performed in a dedicated environment that is similar to production. The software artifacts deployed here originate from the automated build system and are identical to those which would later be deployed to the production system, except that they are configured for the specific test environment regarding the database system, utilized web services, and the user control list. The manual and automated tests that are performed in this environment are focused on evaluating all use cases and test cases. Once an application has passed all tests in the integration test environment, it can then be promoted to the next environment.

3.6.3 System testing

The QA environment is set up to be identical to the production environment, with the exception of configuration. The artifacts deployed are those which previously passed integration testing. During QA, any previously identified bugs or enhancements are first tested in order to ensure that the code changes fully addressed these. Test cases are executed, recorded and cross-referenced against the requirements matrix. Additional testing in this environment includes stress

and penetration testing. Finally, a full regression testing is performed, including end to end system testing of all use cases using multiple test scenarios. Builds which pass 100% of tests in this environment become release candidates.

3.6.4 User acceptance testing

The user acceptance testing (UAT) process is critical to the deployment of the pharmacy POS system into the production environment. The POS Team will assist the Department in developing test scenarios, build the test cases, provide the setting (facilities and infrastructure), and tools to perform UAT. In addition, UAT supports the independent readiness assessment and builds CMS certification documentation. During UAT, the Department controls the testing specifications with the support of the POS Team who provide use cases and appropriate test data, along with the necessary support staff, technical expertise and tools required to successfully complete and report on UAT activities.

3.6.5 Stress testing

For stress testing, the POS Team uses tools (e.g., JMeter tool) to simulate much larger than normal user loads over an extended time period. Statistics from the testing, such as the total number of runs per use case, average response times, standard deviation, transactions per second, CPU usage, memory usage, network traffic, and other relevant server and network traffic information, are captured and analyzed.

3.6.6 Parallel testing

For parallel testing, the POS Team employs an automated claims test-processing tool that captures real-time claims from the the pharmacy telecommunication switching companies. These captured claims are used to build test files to perform real time processing simulations, and provide us with the ability to perform “what if” scenarios based on benefit plan changes, test quality assurance, and performance and regression testing.

3.7 Phase closeout project reviews

The POS Team will meet with the Department to review deliverables provided within a defined project phase. If any lessons learned can be captured, they will be documented in the meeting notes and used to improve future deliverables. Once all of the deliverables for the phase have been approved, the POS Team will create an invoice and reference those items that were delivered and approved within the phase, including the RFP citation number.

3.8 Continuing support

We are proud of the first-class work that our highly skilled and experienced employees consistently produce in conjunction with our state-of-the-art technology investments. In support of these superior results, and to ensure that all contractual obligations are being met to the satisfaction of the contracting entity, operations and administrative staff have collaborated to develop quality control and assurance strategies to provide monitoring of all operations related to a given contract. Additionally, these methodologies are documented and available for reference or for use in training new staff.

Throughout the life of our contracts, we continuously measure our performance, identify opportunities for improvement based on analysis of established metrics, and then implement changes in our activities designed to achieve measurable enhancements. Our commitment to continuous quality advancement is the driving force behind all work we perform for Medicaid agencies, and this principle will be rigorously applied to the work completed for the Department.

4 Tools, techniques and methods

4.1 Deliverable quality assurance

All POS deliverables and artifacts will follow the Deliverable Review Methodology and Standards as established by the Quality Assurance/Quality Control (QA/QC) Team.

The POS Team will conduct a walkthrough of each deliverable at various stages during the development of documents and systems, work closely with the Department to define the format and content for each, and then strictly adhere to the Department's process for reviewing and approving the deliverables.

A complete deliverable review cycle takes up to 20 business days consisting of the following proceedings:

- Ten business days are allocated for the initial deliverable review by the State Management Team, Project Director, Quality Project Director, and QA/QC Team
- Five business days are then provided for the POS Team to address comments from the initial review
- The review cycle then concludes with five business days being set aside for the review team to confirm the changes and resolve any remaining unresolved comments

Unresolved comments remaining at the end of the 20-day review period are escalated to the Project Director and Quality Project Director.

The deliverable review process is discussed in detail in the Deliverable Review Methodology and Standards, which is located in the Project Library.

4.2 Document repository

GHS will utilize the Egnyte® document storage platform that is readily accessible through any web-based browser (also referred to as “Project Portal”).