

State of Vermont

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Agency of Digital Services EPMO

Quality Management Plan

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prepared By: (Your Name)

Date of Publication: mm/dd/yyyy

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Table of Contents

Table of Contents 2

Revision History 3

Quality Management Approach 4

Agile Quality Characteristics 4

Quality Best Practices 5

Quality Tools 5

Quality Management Roles and Responsibilities 6

Quality Assurance, Measurement and Control Processes 7

Quality Assurance 7

Completeness and Correctness 7

Definition of Ready / Definition of Done 8

Acceptance Criteria 8

Contractor Quality Management 8

Deliverables Management 10

Deliverables - Documentation Quality 10

Deliverables - Code Quality 10

Quality Standards 10

Quality Control 11

Quality Control Metrics 12

Procedures 16

ACCEPTANCE OF SERVICES 16

Quality Measurement 17

Quality Management Metrics 17

**Project Management-related Quality Metrics** 18

**Project Deliverables-Related Metrics** 19

Testing Exit/Entry Criteria 20

Testing standards 25

Defect Management 26

Appendix A: Example Quality Assurance Surveillance Plan 29

INTRODUCTION 29

STANDARD 29

PERFORMANCE REQUIREMENTS MATRIX 29

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author(s) | Revision Notes |
| 1.0 | 6/14/2022 | Stacy Gibson-Grandfield | Initial Draft |
| 2.0 | 6/28/2022 | Helen Tanona | Version 2.0 |
| 3.0 | 7/6/2022 | Josiah Raiche | Version 3.0 |
| 4.0 | 8/5/2022 | Stacy Gibson-Grandfield | Version 4.0 |
| 5.0 | 11/14/2022 | Stacy Gibson-Grandfield | Version 5.0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Following ISO 9004:2018, Quality Management is key for guiding the State of Vermont to continued success during IT implementations. Quality Management processes provide the documents and checklists to effectively manage project and product quality from the Initiation Phase through Maintenance and Operations (M&O). A Quality Management Plan (QMP) is created during the Planning Phase of the project, traditionally as a sub-plan to the Project Management Plan (PMP) but may serve as a stand-alone artifact of a project. The QMP’s intended audience is the Project Manager, Project Team, Project Sponsor, and any senior leaders whose support is needed to carry out the plan. The QMP outlines the Quality Management Practices, Metrics, and Tools that the project will use to ensure a successful outcome.

The goal of quality management is to ensure that services and products are of sufficient quality to meet the needs of the State. Quality is defined by the degree to which the customer’s requirements are met. All projects are expected to result in reliably working software/product, improved business processes for supporting business operations, and additional reports for measuring overall system performance.

# Quality Management Approach

Quality Management is an integral component of the management approach. It is interwoven with all aspects of the project lifecycle and the methods used to produce the solution. It contributes to the transparency of the development and operations methodologies by providing stakeholder insight into the quality of the solution, in addition to other project deliverables and project processes.

The Quality Management Plan aims to provide repeatable and consistent results. A quality process imposes discipline on the work of the project team, improving the ability of each team member to produce high quality work products and deliverables. Quality management is a key component to a successful implementation.

## Agile Quality Characteristics

The State of Vermont has adopted a hybrid agile approach to project management. In contrast to traditional end-of-project testing cycles, in agile implementations testing happens throughout the project. Agile requires regular collaboration between testers, developers, and Business Analysts (BA’s) through daily stand-ups, sprint reviews, and retrospectives. This collaboration ensures quality is considered from the beginning.

When defects are identified, teams eliminate them at their source by not only fixing the defect but adjusting their process to improve reliability. Teams avoid creating an extensive backlog of defects to prioritize against features.

## Quality Best Practices

* Quality is the responsibility of the entire team.
* Quality is proactive and built into the product.
* Agile teams test early and often.
* Teams should leverage at development strategies such as Test-Driven Development, Behavior-Driven Development, and pair programming.
* Testers should be writing their test scripts the first day of a sprint, working with developers from the start. Waiting too long into a sprint before testing risks the team not completing their work as testing activities run out of time.
* User Acceptance Testing should occur in small batches regularly. Accepted features should have their tests automated.

# Quality Tools

The State of Vermont has a suite of tools that will be used for quality management purposes. Each project will incorporate use of these tools as appropriate, guided by the Requirements Management Plan, Test Plan and input from the ADS Quality Management team.

* Requirements and User Acceptance Testing
  + Azure DevOps is the state of Vermont’s preferred tool that supports teams to plan and capture requirements, development information, and testing information. It provides traceability all through the development cycle. The detailed requirements/user stories and the success of testing processes will be important in determining if the level of quality reached is acceptable.
* Code Quality
  + CodeScan.io is a platform for continuous inspection of code quality and security based on SonarQube. The State’s SonarQube instance is used for all Salesforce projects and includes a suite of custom rules to ensure that apps will work together well in the shared environment.
  + SonarQube is an open-source platform that provides automatic code review for bugs, vulnerabilities, and code smells.
* Regression Testing
  + Testim.io is the State’s preferred platform for automated regression testing. It offers scheduled and on-demand test capabilities for web applications.

Additional tools may be relevant in specific cases, such as linters, DAST Scanners, and load testing tools. This list is meant to establish a baseline, not limit the use of tools to improve quality.

# Quality Management Roles and Responsibilities

The role of the QM team is to assist the technical staff to continually improve the quality of their work products and services. The QM team is responsible for establishing processes and procedures that accurately verify and validate the adherence of deliverables to applicable standards, guidelines, and procedures. Resourcing for these roles will vary depending on the dynamic and technologies being leveraged for the project. The team should consider and decide the most effective way to fulfill these roles.

The QM team will be involved at the start of the project. They will participate in the development of the project plan to establish their function within the project and to provide input into the project’s schedule to ensure that QM activities are identified, and that time is allotted for QM activities. Funding for the QM team members will be planned within the task hours and cost structure.

The organizational responsibilities as they relate to QM are:

|  |  |
| --- | --- |
| Role | Quality Management Responsibilities |
| Project Manager | * Manages overall project performance. * Ensures QM activities are conducted * Ensures compliance with the QM program * Ensures responses to deficiency reports from QM reviews and audits |
| Project Quality Reviewer | * Representative of the Quality Management team * Conducts deliverable reviews * Ensures work products adhere to appropriate standards * Develops audit and review procedures for project activities * Ensures the QM processes and procedures adequately control project quality * Ensures the QM activities accurately measure the product, service, and process quality * Reviews and approves specified deliverables for release to customer * Promptly reports results of audits to the project manager and program manager * Provides recommendations to improve product quality |
| Project Sponsor | * Provides management support, supervision, and oversight for the QM function * Ensures the independence of the QM function * Makes available staff and other resources as needed to support QM * Ensures resolution of concerns, problems, and issues * Reviews QM audits and reports |
| Project Team Members | * Recommends changes in procedures to improve processes * Implements task level quality control based on QM standards, policies, and procedures * Participates in reviews and audits * Performs corrective actions or process improvements in response to QM findings * Manages and controls defects/errors and corrections * Tracks the status of defects/errors until closed |

The effectiveness of the QM team’s effort depends on the support and commitment of the technical staff and all levels of management. All affected groups should be trained in the principles of Quality Management and be committed to the proper inclusion and performance of QM activities within their work efforts.

# Quality Assurance, Measurement and Control Processes

Quality Management starts with establishing quality standards for what is measured and the target result. Quality assurance (QA) activities focus on the processes being used to manage and deliver the solution in line with these quality standards and can be performed by a Manager, business sponsor, or a third-party reviewer. Quality control (QC) activities are performed continually throughout a project to verify that project management and project deliverables are of high quality and conforming to the expected standards.

Quality Assurance provides a leading indicator of the likelihood of delivering a quality project. Quality Control provides a lagging indicator of how well deliverables meet the standard, and ensures deliverables are brought into compliance with the standard.

## Quality Assurance

### Completeness and Correctness

The project team and major stakeholders should define criteria for when each major deliverable is complete and correct. The deliverables are then evaluated against these criteria before they are formally approved.

The status of Completeness confirms that the product delivered meets the functional and non-functional requirements for the solution.

### Definition of Ready / Definition of Done

The Definition of Ready (DoR) defines criteria a story must meet before being taken into a sprint. The value a good DoR provides is it reduces the possibility of errors and defects due to misunderstandings and/or ambiguity in the requirements.

A Good Definition of Ready:

* Provides a quality gate to protect the team from poorly formed user stories by the business
* Reduces the possibility of errors and defects due to misunderstandings and/or ambiguity in the requirements
* Minimizes wasted time in trying to clear up poorly defined or incompletely defined requirements in the middle of a sprint

The DoR varies by project and team and evolves as the dynamics of the project shift. Teams should review often throughout the project lifecycle to establish a definition that works for the team.

A clear Definition of Done (DoD) drives the level of quality required to produce a potentially shippable product. Each team should have a DoD that is comprehensive to the entire team, for example, a feature is done once it has passed both Integration Testing and User Acceptance Testing. The DoD sets a level of quality that must be met.

### Acceptance Criteria

Each User Story should have clear Acceptance Criteria. Acceptance Criteria is a set of conditions that must be met before the customer or stakeholder accepts a Feature or User Story. Acceptance criteria is the result of a conversation between the product owner and the development team. In addition, acceptance criteria:

* Captures business criteria that must be met for a user story to be complete.
* Represents the quality requirements expected by the customer.
* Specifies the conditions under which a particular user story is fulfilled

The acceptance criteria are closely related to the Definitions of Ready and Done. The team should review acceptance criteria during agile ceremonies and adjust the Definitions of Ready and Done to ensure future acceptance criteria are well-defined.

### Contractor Quality Management

The Contractor is responsible for management and quality control actions to meet the terms of the contract and quality standards set by the State. The role of the Product Owner (PO) and Vermont Product Team is quality assurance to ensure contract standards are achieved.

The Contractor shall perform all work required in a satisfactory manner in accordance with the requirements of the contract. The Contractor shall notify the Product Owner for appropriate action if it is likely that the Contractor will not achieve successful final delivery of the software code in accordance with the performance objectives and acceptable quality levels.

The Contractor will be responsible for working with the State of Vermont and conducting requirements planning sessions to develop detailed system functional and non-functional requirements, as well as a testing plan that traces back to each requirement using a requirements traceability matrix and Azure DevOps. These expectations are further delineated in the project’s Requirement Management Plan and Test Plan.

Contractor will be expected to develop the solution to meet the acceptance criteria as defined in the requirements/user stories and provide a Testing Plan that describes the processes they will use to verify the quality of the product. The activities conveyed in the Testing Plan must adequately demonstrate that technical requirements have been met, and accordingly that the quality of the solution functionality and business processes meets expectations.

The Contractor’s solution will conform to ISO *9001:2015 that will:*

1. *demonstrate its ability to consistently provide products and services that meet customer and applicable statutory and regulatory requirements, and*
2. *enhance customer satisfaction through the effective application of the system, including processes for improvement of the system and the assurance of conformity to customer and applicable statutory and regulatory requirements. -* [*https://www.iso.org/standard/62085.html*](https://www.iso.org/standard/62085.html)

For a System Testing Plan that is all-encompassing for a project, viewing it as a Master Test Plan, to include Sub-Test Plans for the different Test Levels (Unit, SIT, UAT, etc.) then using just the ISO/IEC/IEEE 29119 (parts 2 and 3) would be sufficient and cover most of what is required.

In addition, Contractors will conform to standards set by industry principles and standards, and sufficiently address the challenges represented within a multi-Vendor, integrated systems solution**.**

The State of Vermont has adopted a Quality Assurance Surveillance Plan to provide confidence that all quality requirements related to code have been met. An example of the QASP, demonstrating key areas of quality and the acceptable quality levels, is included as an Appendix to this Quality Management Plan.

# Deliverables Management

## Deliverables - Documentation Quality

For documentation deliverables, the project team will use the State’s pre-defined Deliverable Review and Acceptance as described in the contract or as agreed to by the State. All deliverables should be tracked in a Deliverables Tracker document or Project Management tool, with progress reporting shown as part of the Project Status Meetings. Contractor internal reviews should be held to confirm that any drafted deliverables meeting technical writing standards in addition to the Deliverable Expectation Document prior to being presented to the State.

Documentation Quality is defined as all comments from the State review having been addressed appropriately. In some cases, a post-review walkthrough may be required for the author to explain why certain review comments cannot be addressed. Throughout the life of the project, frequent reviews and continuing education should be provided to ensure both compliance to, and an appropriate level of quality, is present in all project artifacts and documentation upon finalization and publishing.

## Deliverables - Code Quality

Contractor will use Coding Standards covering all programming languages used in the system to define code quality. Contractor will perform internal code reviews on 90% of code produced for this project to confirm adherence to the Coding Standards as defined by the State. These reviews should (at a minimum) concentrate on areas of critical functionality.

Code Quality is defined as confirming the coding capacity will be able to address the backlog in the required schedule, and all the QASP Performance Standards met.

## Quality Standards

Quality standards are requirements, specifications, guidelines, or characteristics that can be used consistently to ensure that materials, products, processes, and services are fit for their purpose. Standards provide organizations with the shared vision, understanding, procedures, and vocabulary needed to meet the expectations of their stakeholders.

Quality standards apply to all participants of the project (both contractor and State) and to all deliverables (including code, hardware, and other project deliverables) and processes applied within the project.

# 

# Quality Control

The basic ceremonies in a typical agile project provide an opportunity to continually review and control quality. Standard ceremonies include Release Planning, Sprint Planning, Daily Standups, Sprint Reviews, and Sprint Retrospectives. Each ceremony provides an opportunity to validate and verify quality:

*Release Planning* – Defining the release features allows opportunities for quality definition. The business has an opportunity to define the level of quality expected regarding functionality, business value, performance, and many other attributes (I.e. definition of done).

*Sprint Planning* – The team will discuss how to build functionality to meet acceptance criteria.

*Daily Stand-ups* – Each daily stand-up provides an opportunity to check quality daily.

*Sprint Reviews* – During sprint review, the users and stakeholders can inspect what the team has built, giving another opportunity to ensure the functionality provides value.

*Sprint Retrospectives* – During sprint retrospectives, the team reviews their processes and how they worked as a team. It provides a golden opportunity to improve their processes, team interaction, and quality. This is a good time to find any circumstances that led to defects, and then brainstorm how to remove those causes so they don’t happen again during the next iteration.

In addition to the quality control practices described above, projects may also conduct additional activities as described below according to their project needs:

Peer Reviews – These reviews enable deliverable and work product deficiencies to be identified and corrected early in the project.

Deliverable Reviews – These reviews focus on formal project deliverables defined in the project schedule.

Periodic Assessments – These assessments focus on trends or changes in project processes and are a routine part of project meetings, project status reports, and many other project communication events.

Performance Measurement – The overall goal of performance measurement is to provide quick feedback to the Project Team and to rate the project’s performance against the chosen targets and applicable standards.

Continuous Improvement – Quality assurance activities provide the opportunity to identify, evaluate, and implement improvements to project processes on a continuous basis. The goal is to achieve solutions and services that meet or exceed the project requirements and satisfy the project goals.

# Quality Control Metrics

The State of Vermont will evaluate the performance objectives reflected below by reviews and acceptance of work products and services. As indicated, the State will assess progress towards the final delivered software code. Note that the performance requirements listed below are required for the final deliverables. However, the sprints and incremental delivery of code will be assessed by the State to ensure that the contractor is on a path to successful final delivery.

|  |  |  |  |
| --- | --- | --- | --- |
| Tested Code | Code delivered under the contract must have substantial test code coverage and a clean code base  Version-controlled Vermont GitHub repository of code that comprises product that will remain in the public domain | Minimum of 90% test coverage of all code | Combination of manual review and automated testing |
| Properly Styled Code | [GSA 18F Front End Guide](https://frontend.18f.gov/#js-style) | 0 linting errors and 0 warnings | Combination of manual review and automated testing |
| Accessible | Web Content Accessibility Guidelines 2.1 AA (WCAG 2.1 AA) standards | Web Content Accessibility Guidelines 2.1 AA (WCAG 2.1 AA) standards | <https://github.com/pa11y/pa11y> |
| Deployed | Code must successfully build and deploy into staging environment and must be compatible with data schemas used in production. If data schemas are not available, code must successfully build and deploy into production environment | Successful build with a single command | Combination of manual review and automated testing |
| Documentation | All dependencies are listed and the licenses are documented. Major functionality in the software/source code is documented | Individual methods are documented inline using comments that permit the use of tools such as JsDoc. System diagram is provided | Combination of manual review and automated testing, if available |
| Secure | OWASP Application Security Verification Standard 3.0 and meet the requirements of an application in a CMS MARS-E compliant environment | Code submitted must be free of medium- and high-level static and dynamic security vulnerabilities | Clean tests from a static testing SaaS (such as Veracode or Snyk) and from OWASP ZAP, along with documentation explaining any false positives |
| Security | Vulnerabilities (vulnerabilities)   * Number of vulnerability issues. * Number of new vulnerability issues.   Security remediation effort   * Effort to fix all vulnerability issues. The measure is stored in minutes in the DB. An 8-hour day is assumed when values are shown in days. * Security remediation effort on new code same as Security remediation effort but on the code changed on New Code.   Security Hotspots   * Number of Security Hotspots   Security Hotspots on new code   * Number of new Security Hotspots on New Code.   Meets other Application Security Standards   * OWASP Application Security Verification Standard 3. * CMS MARS-E compliant environment. | Security Rating  A = 0 Vulnerabilities  B = at least 1 Minor Vulnerability  C = at least 1 Major Vulnerability  D = at least 1 Critical Vulnerability  E = at least 1 Blocker Vulnerability  Code submitted must be free of medium- and high-level static and dynamic security vulnerabilities. | Clean tests from a static testing SaaS, along with documentation explaining any false positives |
| Security Review | Security Review Rating on new code   * Security Hotspots Reviewed * Percentage of Reviewed (Fixed or Safe) Security Hotspots. * Ratio Formula: Number of Reviewed (Fixed or Safe) Hotspots x 100 / (To Review Hotspots + Reviewed Hotspots) * New Security Hotspots Reviewed * Percentage of Reviewed Security Hotspots (Fixed or Safe) on New Code. | The Security Review Rating is a letter grade based on the percentage of Reviewed (Fixed or Safe) Security Hotspots.  A = >= 80% B = >= 70% and <80% C = >= 50% and <70% D = >= 30% and <50% E = < 30% |  |
| Maintainability | (Formerly the SQALE rating.) Rating given to your project related to the value of your Technical Debt Ratio.  Code Smells   * Total count of Code Smell issues.   New Code Smells   * Total count of Code Smell issues raised for the first time on New Code.   Technical Debt   * Effort to fix all Code Smells. The measure is stored in minutes in the database. An 8-hour day is assumed when values are shown in days.   Technical Debt on New Code   * Effort to fix all Code Smells raised for the first time on New Code.   Technical Debt Ratio   * Ratio between the cost to develop the software and the cost to fix it. The Technical Debt Ratio formula is:   Remediation cost / Development cost   * Which can be restated as: * Remediation cost / (Cost to develop 1 line of code \* Number of lines of code) * The value of the cost to develop a line of code is 0.06 days.   Technical Debt Ratio on New Code   * Ratio between the cost to develop the code changed on New Code and the cost of the issues linked to it. | The default Maintainability Rating grid is:  A=0-0.05, B=0.06-0.1, C=0.11-0.20, D=0.21-0.5, E=0.51-1  The Maintainability Rating scale can be alternately stated by saying that if the outstanding remediation cost is:  <=5% of the time that has already gone into the application, the rating is A  between 6 to 10% the rating is a B  between 11 to 20% the rating is a C  between 21 to 50% the rating is a D  anything over 50% is an E |  |
| Reliability | Bugs (bugs)   * Number of bug issues.   New Bugs   * Number of new bug issues.   Reliability remediation effort   * Effort to fix all bug issues. The measure is stored in minutes in the DB. An 8-hour day is assumed when values are shown in days.   Reliability remediation effort on new code   * Same as Reliability remediation effort but on the code changed on New Code. | Reliability Rating (reliability rating)  A = 0 Bugs  B = at least 1 Minor Bug  C = at least 1 Major Bug  D = at least 1 Critical Bug  E = at least 1 Blocker Bug |  |
| User Research and Design Artifacts | Design research and usability testing activities must be conducted at regular intervals throughout the development process (not just at the beginning or end) to ensure the user needs are well understood and that design solutions work well for users | During the first sprint, vendor shall establish a design research plan in collaboration with the Product Team  This plan must account for the availability of resources, articulation of research methods, and delivery of research-related records.  In subsequent sprints, research-related records will be delivered in accordance with the design research plan | Cross-reference research-related records with other project documentation to ensure that research is properly accounted for and communicated. |

# 

# Procedures

Delivery of all software assets will occur by pull request from the Contractor’s repository to the appropriate State repository. If inspection results are satisfactory, the pull request will be merged; otherwise, deficiencies will be noted in the pull request or through issues as described below. The State Product Team and Product Owner may find the delivery satisfactory even though further work is required in the next sprint.

At the conclusion of each sprint, the State Product Team will review the related functionality to ensure compliance with acceptance criteria and requirements of the user stories. All clarifications and changes to the user stories during the sprint that are agreed upon are documented in the issue tracker. Incomplete or inadequate code and user stories will be noted in a mutually agreed-upon issue tracker with links to each issue shared with the Product Owner. The Contractor may respond in that tracker as appropriate, addressing the accuracy and validity of the defect as well as any planned corrective action (if not already noted). The issue tracker will be updated as revised acceptance criteria are added to the incomplete backlog items as part of the backlog grooming process. The Contractor’s team will discuss and document actions to prevent recurrence in the sprint retrospectives.

At the conclusion of the period of performance, the State Product Team will follow a similar procedure to document discrepancies and to assess overall performance.

## ACCEPTANCE OF SERVICES

The Product Owner (PO) will review all work products for compliance with performance standards described in the Scope of Work and monitoring procedures described in this QASP. The Product Owner will not accept work products for the contract until all defects have been corrected.

Other Quality Assurance activities include continuous review and identification of areas for improvement through project meetings. Metrics and reports on quality adherence should be made available to the project stakeholders to correct downward trends in quality. Conversely, upward trends in quality should be analyzed to determine best practices that can be used elsewhere.

## Quality Measurement

Metricsrelated to QA and development are relevant to a project’s success. Identifying and defining key quality metrics should occur early in the project to ensure that success can be measured and tracked throughout the project. Below is a list of some of the more common metrics that may be identified as key quality indicators.

**Metrics - Development**

* Lead Time
* Cycle Time
* Team Velocity (Points per sprint/day)
* Defect Resolution Time
* Active Days (vs. Inactive Time on Administrative tasks)

**Metrics – Testing**

* Defect Open/Close Rates
* Burndown Chart
* Defect Category Rates
* Defect Detection Percentage (DDP)

The following checklist provides a basis to develop key quality indicators for a project. The checklist questions may not be applicable to every project, and each project may adapt these questions into metrics as needed.

### Quality Management Metrics

The QM team will work with the technical support staff to identify indicators and their associated measures (Metrics) that are needed to control performance and predict future status of processes used to produce products and services. The metrics will be used to help determine when and where a problem is occurring and what type of impact it will have on the product or service. The metrics will be used to base decisions concerning the selection of best practices to implement in the project.

Metrics that are necessary to monitor the effectiveness of QM processes and procedures are:

1. Number of reviews (QM activities) conducted

2. Status of non-conformance items identified

3. Status of action items open/closed/on-hold

4. Number of days to correct and close a non-conformance item

5. Customer satisfaction levels relating to product and service quality

6. Trends for process improvement

7. Lessons learned

### **Project Management-related Quality Metrics**

|  |  |  |  |
| --- | --- | --- | --- |
| **At the Beginning of the Project** | **Y N n/a** | | |
| * Have the project sponsor and key stakeholders been identified? * Did the key stakeholders participate in the planning? * Have the sponsor(s) and major stakeholders formally approved the Project Definition? * Are the objectives, scope and deliverables clearly defined? * Are the resource requirements adequate? * Do the team members understand the time required and the deliverables they are responsible for? * Have the managers of team members that are not allocated full time to your team agreed to have the people available when you need them? * Do the estimated effort, cost and duration appear reasonable? * Does a project workplan exist? * Does the workplan contain at least two months of detailed work activities? * Does the workplan contain milestones that can be used for follow-up QA checks? * Does the workplan contain a full testing process, including user acceptance testing? * Do Project Management Procedures exist to define how the project will be managed? |  |  |  |
| **At Every Major Milestone** | **Y N n/a** | | |
| * Is the workplan being used to manage the work performed by the team? * Does the workplan accurately reflect the remaining work effort? * Does the workplan contain detailed work activities for at least the next two months? * Can the project manager clearly explain where the project is vs. where it should be at this time? * Will all the deliverables specified in the Project Definition be completed? * Are the project finances being actively managed to complete within the budget? * Is the project on track in terms of cost, duration, and quality? * Are old risks being managed and new potential risks being identified? * Are issues being addressed and resolved in a timely manner? * Are scope change requests being properly identified and managed? * Were any major changes encountered that require that the Project Definition be updated? * Are status reports and status meetings being utilized? * Are major stakeholders being communicated to effectively? * Are the business customers happy with the project progress so far? * Are customer expectations being properly managed? * Are proper processes being utilized to ensure completeness, correctness, and overall quality? |  |  |  |

### **Project Deliverables-Related Metrics**

|  |  |  |  |
| --- | --- | --- | --- |
| **Analysis Milestone** | **Y N n/a** | | |
| * Did the project complete the appropriate deliverables? (Examples include Business Requirements Report, Conceptual System Design Report, Testing Strategy, Data Conversion Strategy and Training Strategy.) * Were the appropriate documents approved by the appropriate customer or project sponsor? * Is the customer engaged on the project to the degree required? * Is the project following appropriate company standards, guidelines, and policies? |  |  |  |
| **Design, Program, Test Milestones** | **Y N n/a** | | |
| * Did the project complete the appropriate deliverables? (Examples include Business System Design Report, Technical System Design Report, Testing Plan and Data Conversion Plan.) * If the preceding deliverables are not created, discuss how the testing was accomplished, how the training will be performed and how data will be converted. * Is the project following appropriate company standards, guidelines, and policies? * Is the project following the standard company technology architecture? * Did the project documents go through an appropriate internal team review and approval process? * Were the appropriate documents approved by the appropriate customer or project sponsor? * Was a full testing process used to ensure the solution is ready to be implemented (unit, integration, system, and acceptance testing)? * Are appropriate plans in place for disaster recovery, security, and training? * Have all the various teams that may be involved with the implementation and deployment of the application been contacted, and do they know what is expected from them? |  |  |  |
| **Implementation Milestone** | **Y N n/a** | | |
| * Has the production application been approved and accepted as completed by the Project Sponsor? * Were the technical components reviewed/inspected by qualified technical associates prior to installation? * Is the production application being monitored, and are problems being corrected as necessary? * Are initial problems being resolved in a timely manner? * Has a process been defined and communicated to transition the production application to the support organization? * Has all the appropriate solution documentation been completed (system, user, support, etc.)? |  |  |  |

### Testing Exit/Entry Criteria

#### Unit Testing / Configuration Testing Exit Criteria

The planned level of unit testing in scope consists of informal unit and smoke testing to the extent possible in the development environment. Unit test cases and execution results are anticipated to be documented, managed, and reported on via designated testing tool(s)

The following criteria must be met to exit the Unit Testing phase:

* Requirements elicited and documented
* All Unit Test Scenarios have been executed and mapped to requirements
* Defects have been recorded in testing tool and all fixes have been retested & validated
* No outstanding defects unless written approval by the State
* Release notes compiled and sent
* Deployment notes and release entry framework is complete
* RTM is updated with results

Prior to the start of Systems Integration Testing, the following criteria must be met:

* Code deployed to the SIT environment.
* Unit Test Exit Criteria has been met
* Testing scope identified based on approved requirements
* System Integration Test Plan document is approved by both Contractor and SoV. This document is needed for entrance unless written approval by the State is provided.
* Test scenarios and expected results have been created based on scope requirements and process flows by the Contractor, and have been approved by SoV
* Test Scenarios, conditions, scripts and expected results are documented, including connectivity testing, in testing tool where applicable
* All planned Test Cases drafted, reviewed, and approved by both Contractor and SoV in testing tool
* Provisioning complete for any tools/environments to be used by testers
* Environment being used has been updated with the necessary build and/or configuration
* Testing timeline documented and approved as part of the Project Plan
* Testing timeline must be documented in the IMS including IV&V testing where necessary
* RTM is updated and mapped to the test scenarios
* Regression scenarios are exercised after defects found during SIT have been resolved and deployed
* Test Cases mapped to distinct requirements in testing tool

#### SIT Exit Criteria

For exit from SIT, the following exit criteria must be met:

* Test cases in scope for SIT executed to one or more of the following criteria:
  + - * 100% test case execution with acceptable pass rate of 90% or higher with zero critical or high defects
* All high-risk / high-priority test cases identified and agreed up on by the SoV have been executed with passing status
* No critical or high (Severity 1 or Severity 2, respectively) defects open, or written authorization from SoV has been received
* Test cases not previously mapped to requirements are executed and mapped
* All remaining open defects (Severity 3 and Severity 4) identified with SoV-approved plan of action for remediation based on impact analysis
* All open defects are mapped to failed test cases
* All closed defects mapped to passed test cases
* All planned Test Cases, Connectivity, and Regression Suite scenarios have been executed and are in a passed status.
* Any scenarios not in a passed status have been documented and are agreed to not be a blocking issue by Contractor and SoV
* All non-closed defects have either been validated or agreed to be deferred by mutual agreement of Contractor and SoV
* All identified defects have been recorded and linked to test scenarios in testing tool and all fixes have been retested & validated
* Test scenarios in testing tool accurately reflect pass/fail/other status
* RTM is updated with results
* RTM export shows acceptable test coverage of requirements based on test case pass/fail rate

Release Notes / Test Summary Report are updated for each enhancement and code fix and are populated with SIT test results, test date, environment, & tester

#### UAT Entry Criteria (SoV)

Prior to the start of User Acceptance Testing, the following criteria must be met:

* Testing environment built, with code deployed
* Testing scope identified and approved by SoV Business
* Test Cases drafted, reviewed, and approved in testing tool
* Test Cases mapped to distinct requirements in either one-to-one or one-to-many relationships
* Provisioning complete for any tools/environments to be used by testers
* SIT Exit Criteria has been met
* No Critical or High defects exist from previous testing level (SIT)
* All non-Critical defects have been identified and communicated to SoV
* Deliverables such as Test Plan and/or Test Strategy have been reviewed and approved

#### UAT Exit Criteria (SoV)

For exit from UAT, the following exit criteria must be met:

* Test cases in scope for UAT executed to one or more of the following criteria:
  + - * 100% test case execution with acceptable pass rate of 90% or higher with zero critical or high defects
      * All high-risk / high-priority test cases identified and agreed up on by the SoV have been executed with passing status
* No critical or high defects open
* Test cases not previously mapped to requirements have been executed and mapped
* All non-closed defects have either been validated or agreed to be deferred by mutual agreement of Contractor and SoV

Additional criteria for a Go/No-Go decision to deploy into production is required. These criteria vary from project to project, but generally include:

* No outstanding Severity 1 or 2 defects open, or written authorization from SoV has been received
* Review of Severity 3 and Severity 4 defects by SoV with an approval to proceed based on impact analysis
* All defects (open or closed) are mapped to test cases for bidirectional traceability
* All identified defects have been recorded and linked to test scenarios in testing tool, and all fixes have been retested & validated
* Regression scenarios are exercised after defects found during SIT have been resolved and deployed
* Test scenarios in testing tool accurately reflect pass/fail/other status
* Test summary report completed and reviewed
* Project-specific deliverable documents complete and provided to the State
* Approval for deployment by authorizing stakeholders

The following criteria must be met to exit the UAT testing phase unless written approval by the State is provided:

* All planned Test Cases have been executed and have a pass/fail status. Any test cases not in a passed status have been documented and are agreed to not be a blocking issue by Contractor and SoV.

Release Notes / Test Summary Report for each enhancement and code fix and are populated with UAT test results, test date, environment, & tester

##### Standard Levels of Testing typically in scope:

* Unit Testing
* System Integration Testing (SIT)
* User Acceptance Testing (UAT)
* Regression Testing
* Production Testing - also known as User Verification Testing (UVT)
* Security Testing is considered in scope if any new development is performed
* Performance Testing
* Section 508 Compliance testing
* Regression Testing

#### Summary report of SIT and UAT testing progress

Detailed SIT and UAT testing reports such as:

* + Test coverage of requirements
  + Defect status reports, including open defects
  + Test case pass rate
  + Test case status with all pass/fail
* How often is a User Story going through UAT (no more than 2x is stnd).

#### Regression Testing

Regression testing will use the Testim.io (testing tool) where appropriate. Expected results for passing regression testing include:

* 75% user stories have automated test coverage,
* 60% user stories are part of an end-to-end automated test suite,
* 60% user stories have automated boundary testing (point at highest and lowest values and go one higher and lower to make sure they both fail).
* For platforms, should be built using standard VT users and permission sets using our standard permissioned user in the States Testim instance.
* Required on platforms (Salesforce) and anytime any project has a release date and during seasonal/scheduled platform upgrades.
* Requiring reporting of metrics –

## Testing standards

Testing activities should adhere to include ISO/IEC/IEEE 29119. These standards outline the expectations for any SDLC. There are five categories for these standards:

* ISO/IEC 29119-1: Concepts & Definitions
* ISO/IEC 29119-2: Test Processes
* ISO/IEC 29119-3: Test Documentation
* ISO/IEC 29119-4: Test Techniques
* ISO/IEC 29119-5: Keyword Driven Testing

The most used standard for the State of Vermont is:

* ISO/IEC 29119-1: Concepts & Definitions
* ISO/IEC 29119-2: Test Processes
* ISO/IEC 29119-3: Test Documentation
* ISO/IEC 29119-4: Test Techniques

#### High level process for Testing Activities

Identify Feature Sets: Analyze test basis to understand the test item requirements, combine test features into test set and prioritize by risk value, document the feature set, and communicate it to stakeholders.

Identify Test Conditions: Determine the test condition for each test case; prioritize the best condition based on risk, record test condition in test design specification.

Test Coverage Items: Prioritize test coverage items based on risk, record test coverage items in test design specification. Record the traceability.

Test Cases: Determine pre-condition and input values for one or more test cases and expected result prioritize by risk value, record test cases items in test design specification. Record the traceability, get approval of stakeholder.

Assemble Test Sets: Distribute test cases into one or more test sets based on constraints and execution, record test case in procedure specification, record the traceability.

Test Procedures: Derive test procedures from ordered test set based on pre-condition, postcondition and dependencies, Identify excluded test data, prioritize test procedures based on risk, record test procedures in procedure specification, record the traceability, get approval of stakeholder.

#### Deliverables Expected for Testing

Not all the following deliverables will be expected for all deployments, however, below is a list of all test deliverables and the sequence by which they arrive for any given DDI / M&O deployment.

1. Test Policy – high level and often internal to the testing teams
2. Organizational Test Strategy – internal to testing teams, provides a ‘master’ strategy for testers to refer to for consistency and standardization
3. Test Plan
4. Test Cases
5. Test Data (if required)
6. Test Status Report (Routine updates during active testing)
7. Test Summary Report (Completion of testing)
8. Test Incident Report / Defect Management Log

The Test Plan, Test Cases, Test Status Report, Test Summary Report, and Test Incident Report/Defect Management Log are considered critical deliverables for a Go/No-Go decision

### Defect Management

##### Problem Reporting Procedures

Errors, defects, issues, deviations, and noncompliance items identified in the activities must be itemized, documented, tracked to closure, and reported by the QM team. The QM team must verify all problems were tracked to closure and must provide continuing feedback to management and the technical support team concerning the status of the problem. The QM team will use HPALM to track errors, defects, issues, deviations, and noncompliance items. Items will be submitted to the state in both the daily status report and the final test report (if anything was not fixed).

##### Non-compliance Reporting Procedures

Problems are resolved with the direct producer, when possible.

Problems that cannot be resolved with the technical team are elevated to the project manager.

Problems that have been referred to the project manager are reviewed weekly until they are resolved. Items that cannot be resolved by the project manager within six weeks are elevated to ICS for resolution.

Each phase of testing will include defect reporting, and the communication / escalation processes will be further defined on a deployment-by-deployment basis (different business owners to report to, etc.). The processes will include defining the anticipated resolution time for defects found based on defect level, when a defect can be closed, and how all defects will be tracked respective of testing level.

##### Severity vs Priority

The severity of a bug is derived based on the effect of that bug on the system. It indicates the level of threat that a bug can affect the system such as how much of the application is affected, the impact to end-users, the scope of impact to end-users, operational impact, financial impact, or security concerns.

Priority is set by the business, driven by business-related need and determines how quickly a defect is required to be resolved. Factors driving priority include communications to end-users, compliance issues, leadership initiatives, or level of effort versus cost.

##### Incident vs Defect

Clear language around incident vs defect will need to be determined. These terms can be used interchangeably, or one can represent a technical issue (incident equating to a security breach, infrastructure outage) while the other a code/configuration issue (defect equating to failed requirement).

##### Defect Severity Definitions

1. Severity Level 1 Issues

An Incident shall be categorized as a “Severity 1 Issue” if the Incident:

(a) Renders a business-critical system, service, software, equipment, or network component un-available (complete loss of service), substantially un-available or seriously impacts normal business operations, in each case prohibiting the execution of productive work, and

(b) Affects either

(i) a group or groups of people, **or**

(ii) a single individual performing a critical business function, and

(c) No workaround is possible.

1. Severity Level 2 Issues

An Incident shall be categorized as a “Severity 2 Issue” if the Incident:

1. Does not render a business-critical system, service, software, equipment, or network component un-available or substantially un-available (service is degraded), but a function or functions are not available, substantially available or functioning as they should, in each case prohibiting the execution of productive work, and
2. Affects either
3. A group or groups of people, **or**

ii. A single individual performing a critical business function, and

1. A workaround may be possible
2. Severity Level 3 Issues

An Incident shall be categorized as a “Severity 3 Issue” if the Incident:

1. Causes a group or limited number of individuals to experience an Incident with accessing or using a System, Service, Software, Equipment or network component or a key feature thereof, and
2. A reasonable workaround is not available, but does not prohibit the execution of productive work
3. Severity Level 4 Issues

An Incident shall be categorized as a “Severity 4 Issue” if the Incident:

* 1. Causes a single individual to experience an Incident with access or using a System, Service, Software, Equipment or network component or a key feature thereof,
  2. A reasonable workaround is available, **or**
  3. The Incident is a cosmetic issue, not impacting the execution of productive work.

##### Incident Priority Definitions

Priority is set by the business owner, as it is viewed through the lens of importance and urgency to the business, and the impact it may have on the end users or business processes. Below are some casual definitions around priority.

* **Priority 1:**  **Critical** - Defect should receive immediate effort to fix and deploy to the UAT environment for SoV testing.
* **Priority 2**: **High** - Defect's prompt resolution is important and (*should be)* prioritized for next available sprint or development cycle.
* **Priority 3:** **Medium** - Defect's resolution may be prioritized for a later sprint or development cycle.
* **Priority 4:**  **Low** - Defect's resolution is a nice-to-have.

# Appendix A: Example Quality Assurance Surveillance Plan

## INTRODUCTION

This Quality Assurance Plan (QASP) has been developed to evaluate Contractor actions while implementing the Scope of Work. It is designed to provide an effective method of monitoring Contractor performance for each listed objective on the Performance Requirements Matrix. It also provides a systematic method to evaluate the services the Contractor is required to furnish.

## STANDARD

The Contractor is responsible for management and quality control actions to meet the terms of the contract. The role of the Product Owner (PO) and the State Product Team is quality assurance to ensure contract standards are achieved.

The Contractor shall perform all work required in a satisfactory manner in accordance with the requirements of the contract. The Contractor shall notify the Product Owner for appropriate action if it is likely that the Contractor will not achieve successful final delivery of the software code in accordance with the performance objectives and acceptable quality levels (AQLs) identified below.

## PERFORMANCE REQUIREMENTS MATRIX

The State Product Team will evaluate the performance objectives reflected below by reviews and acceptance of work products and services. As indicated, the State Product Team will assess progress towards the final delivered software code. Note that the performance requirements listed below are required for the final deliverables. However, the sprints and incremental delivery of code will be assessed by the State Product Team to ensure that the Contractor is on a path to successful final delivery.