



**CWDS**  
Child Welfare Digital Services

---

# **Child Welfare Digital Services Project**

## Quality Management Plan

July 2017

## Revision History

Revision / Version #	Date of Release	Author	Summary of Changes
.01	03/16//2015	C. Blehm	Original Document
.02	03/30/2015	C. Blehm	Draft Version of the Quality Management Plan, sent to Review Team for comments and solicitation of feedback.
.03	04/16/2015	C. Blehm	Incorporate feedback and new content into draft Quality Management Plan based on review from IV&V, Nancy and Patti.
.04	05/01/2015	C. Blehm	<p>Based on streamlining recommendations from PMO staff, the Plan was revised to include:</p> <ul style="list-style-type: none"> <li>• Remove Approver and Reviewer History</li> <li>• Move QM Activities from Appendix G to section 1.2 Scope</li> <li>• Move Quality Terms and Acronyms to Appendix A</li> <li>• Remove RACI matrix</li> <li>• Remove section 3.1 Approach and Strategy</li> <li>• Remove all sections identifying quality activities in the DD&amp;I phase                             <ul style="list-style-type: none"> <li>• System Integrator section on Quality Planning</li> <li>• Design and Coding Standards</li> <li>• Testing Standards</li> <li>• System Integrator Deliverables in identifying metrics and measurements</li> <li>• Stage Gate Audits</li> </ul> </li> <li>• Move details of Identifying Quality Targets to new procedure</li> <li>• Move the bulk of the Metrics section to the new Metrics Plan</li> <li>• Move details of Quality Product Reviews to new procedure</li> <li>• Move details of Quality Process Audit to new procedure</li> <li>• Move details of Quality Improvement to new procedure</li> <li>• Move details of Quality checklists to new procedure</li> </ul>
1.0	09/01/2015	C. Blehm	Publish and baseline SDLC Quality Management Plan
1.1	09/14/2016	C. Blehm	<p>Start revision process to transition QMP from SDLC to Agile.</p> <ul style="list-style-type: none"> <li>• Add section for Transition to Agile</li> </ul>

Revision / Version #	Date of Release	Author	Summary of Changes
			<ul style="list-style-type: none"> <li>• Ensure correct usage of CWS-NS versus CWDS</li> <li>• Modify roles and responsibilities for new governance structure.</li> <li>• Address quality management at the service team level.</li> <li>• Identify areas that require more clarification or focus group consensus.</li> </ul>
1.2	10/28/2016	C. Blehm	Align draft QMP to agile based on feedback from focus group. <ul style="list-style-type: none"> <li>• Align Quality Planning section</li> <li>• Align Quality Assurance section</li> <li>• Align Quality Control section</li> <li>• Align Continuous Improvement section</li> <li>• Updated Appendix A</li> </ul>
2.0	05/15/2017	C. Blehm	Annual Update of the Quality Management Plan: <ul style="list-style-type: none"> <li>• Updated Procedure 100: Quality Checklists</li> <li>• Updated procedure 200: Quality Product Review</li> <li>• Updated Procedure 300: Quality Process Audit</li> </ul>
2.1	07/27/2017	C. Blehm	Added in high level summary (executive summary) of this plan and the associated processes. Sent to web team for transition to HTML content for the website.

# Table of Contents

<b>1 EXECUTIVE SUMMARY .....</b>	<b>5</b>
<b>2 INTRODUCTION.....</b>	<b>10</b>
2.1 Purpose .....	10
2.2 Scope.....	11
2.2.1 In-Scope.....	12
2.2.2 Out of Scope.....	13
2.3 Assumptions and Constraints .....	14
2.4 Integration with Quality Management Procedures .....	14
2.5 Document Maintenance .....	14
2.6 Transition to Agile.....	14
2.7 Quality Policy.....	16
2.8 Goals and Objectives of the Quality Program .....	16
2.8.1 Common Vision .....	16
2.8.2 Measurement Criteria .....	16
2.8.2.1 Quality Checklists.....	19
2.8.2.2 Quality Product Review.....	19
2.8.2.3 Quality Process Audit.....	19
2.8.3 Assessment Category Stoplight Status Legend .....	19
2.8.4 CWS-NS Products and Services .....	20
2.8.5 Establishing Quality Processes .....	20
2.8.5.1 Project Management Documentation.....	20
2.8.5.2 Service Team Artifacts.....	20
2.8.6 Establishing Quality Improvement.....	21
<b>3 ROLES AND RESPONSIBILITIES.....</b>	<b>22</b>
<b>4 QUALITY MANAGEMENT APPROACH.....</b>	<b>25</b>
4.1 Determine Approach for Agile Quality Management .....	25
4.1.1 Project Quality .....	26
4.1.2 Product Quality .....	26
4.2 Plan Quality.....	26
4.2.1 Establishing Quality Standards.....	27
4.2.2 Defining Acceptance Criteria .....	27
4.2.3 Agree on the Definition of Done .....	27
4.3 Measure and Assess Quality.....	28
4.3.1 Effective Measurements .....	29
4.3.1.1 Velocity.....	29
4.3.1.2 Hit Rate .....	30

4.3.1.3	Work Remaining: Sprint Burn Down .....	30
4.3.1.4	Work Remaining: Epic and Release Burndown .....	31
4.3.2	Measuring the Quality of Internal Project Documentation .....	32
4.3.3	Measuring the Quality of Development Team Deliverables .....	32
4.3.4	Measuring the Quality of Software Code .....	33
4.3.5	Metrics .....	34
<b>4.4</b>	<b>Quality Assurance .....</b>	<b>34</b>
4.4.1	Quality Assurance Activities .....	35
4.4.1.1	Automated Testing .....	35
4.4.1.2	Burn-Down Charts.....	35
4.4.1.3	Iterative Testing against Acceptance Criteria .....	36
4.4.1.4	Sprint Review .....	36
4.4.1.5	Daily Stand-Up .....	36
4.4.2	Differences between Quality Assurance and Quality Control.....	36
<b>4.5</b>	<b>Quality Control.....</b>	<b>37</b>
4.5.1	Monitoring Controls.....	38
4.5.1.1	Frequent Delivery.....	38
4.5.1.2	Co-Location.....	39
4.5.1.3	Daily Stand-Up Meeting .....	39
4.5.1.4	Sprint Review Meeting .....	40
4.5.1.5	Requirements Traceability .....	41
4.5.1.6	Automated Testing .....	41
4.5.1.7	Releasable Software in every Sprint.....	42
4.5.1.8	Code Reviews or Inspection .....	42
4.5.2	Data to Use in Monitoring .....	42
4.5.2.1	Manage New Risk.....	42
4.5.2.2	Decrease Scope.....	43
4.5.2.3	Increase Scope .....	43
4.5.2.4	Re-Plan the Sprint.....	43
4.5.2.5	Increase Staff or Increase Velocity .....	43
4.5.2.6	Extend the project or add new sprints.....	44
<b>4.6</b>	<b>Improve Quality .....</b>	<b>44</b>
<b>APPENDIX A: TERMS AND DEFINITIONS .....</b>		<b>46</b>

# 1 Executive Summary

Quality Management, in a project context, is concerned with having the right processes to ensure both quality product and a quality project. The organization’s ability to deliver new functionality with the quickest sustainable lead-time and to be able to react to rapidly changing business environments is dependent on the quality of the solution – both the product and the processes. The Agile Manifesto is focused on quality as well:

Continuous attention to technical excellence and good design enhances agility

**Scope.** Quality is managed throughout the project lifecycle of the Child Welfare Services-New System (CWS-NS) Project in the following areas:

- **Quality Planning** – determining a plan for quality by documenting quality standards and framework. In Agile, quality planning tells us what we expect to do.
- **Quality Assurance** – an executing process that is necessary to continuously improve activities and processes to achieve quality. In Agile, quality assurance means ensuring that software is working right. It includes planning activities to demonstrate quality.
- **Quality Control** – a monitoring and control process that ensures every deliverable and work product is measured and tested to ensure results conform to quality standards. In Agile, quality control focuses on monitoring progress though velocity and tracking what software has been incrementally delivered. It involves implementing the plans made during quality assurance activities.
- **Quality Improvement** – adjusting a process for continuous corrective action or process improvement. In Agile, quality improvement is accomplished through sprint retrospectives.

## Quality Activities.

Quality Activity	Purpose	Frequency	Output
Peer Review	A Quality Assurance Internal Review of documents by the owner organization against defined standards.	As per the Project Schedule	Comment Log Matrix as defined in the CWS-Document Management Plan
Sprint Review	A Quality Assurance examination of service team products to verify compliance to standards.	At the end of each Sprint	Sprint Review Presentation or Demonstration
Process Checklists	A Quality Planning Activity where the Quality Manager documents the specific criteria used to evaluate each project management process	As per the Project Schedule	Quality Process Audit Checklist
Quality Process Audit	A Quality Control examination of project management processes, high-level development	As needed	Quality Process Audit Results

Quality Activity	Purpose	Frequency	Output
	processes, and day-to-day practices to verify compliance to project standards.		
Data Analytics Assessment	A Quality Control periodic review and presentation of metrics measurements as prescribed in the data analytics of each service team in Pivotal tracker.	As needed	Metrics Results
Sprint Retrospectives	A Quality Continuous Improvement activity that reviews what went well and what can be improved for each sprint. Issues are identified and the team determines appropriate corrective action or process improvement.	At the end of each Sprint	Sprint Retrospective Results  Could result in a new story in a future sprint
CWDS Digital Service Standards	The CWDS Digital Service Standard establishes the criteria that the CWDS digital services must meet to ensure our services are simple, fast and easy to use. Meeting the criteria means we can consistently provide high quality services and satisfy our users' needs. The Digital Service Standard applies to all CWDS digital services teams.	Used by the development teams during sprint design and development	CWDS Digital Services Standard
CWDS Digital Service Playbook	Builds upon the Digital Services Standard and defines for each standard the description and use, checklist of activities that must be executed and key questions to assess compliance.	Used by the development teams during sprint design and development	CWDS Digital Services Playbook

**Measurement Criteria.** On CWS-NS, we define the specific assessment areas that if measured, will provide a good representation of the basic health and progress of the project. When deficiencies of issues are identified, there must be a clear path to continuous improvement, either in the form of a corrective action or a process improvement. The following assessment areas are evaluated regularly to determine overall project health and ensure that quality is being built into the product in every step:

1. Release Integrity
2. Engineering Practices
3. Agile Processes
4. Governance
5. People and Teams
6. Go Live Readiness

7. Customer Value
8. Technical Debt Management
9. Information Exchange
10. Project Management
11. Technology Management

### **Quality Management Approach.**

**Plan Quality.** Quality Planning involves identifying which organizational standards and which industry standards are relevant to CWS-NS and how to satisfy them. The process outlines the rules that define the quality needs of the project, the required standard for the project's deliverables and processes and how it will be confirmed that the planned requirements are provided in the project's final product.

Agile Project Planning tells us what we expect to do and is usually accomplished in the Sprint Planning activity and throughout the sprint in Backlog Refinement. The job of the Agile Project Manager/Scrum Master is to guide the service team to successful delivery despite the challenges the world throws at the project.

As quality is planned for each service team, the following components are considered:

- Establishing Quality Standards
- Defining Acceptance Criteria
- Agreeing on the Definition of Done

**Measure and Assess Quality.** Measurements communicate values and priorities to the CWS-NS organization. This activity refers to the process of performing quality assurance and conducting quality control activities to assess the actual level of quality of each deliverable and process undertaken within the project. Quality Metrics are parameters or ways to quantitatively assess a project's level of quality, along with the processes to carry out such a measurement. Metrics outline the standard that work will be measured against and are often unique to each project. CWS-NS Quality Metrics will be defined in the planning phase of the project and then measured throughout the project's life to track and assess the project's level of conformity to its established quality baselines.

Agile provides more opportunities to monitor what is going on than traditional methods and hence offers more effective opportunities to intervene (i.e. velocity, hit rate, work remaining on task board, sprint burn-down). The main Agile monitoring technique is to track what software has been incrementally delivered. This is both easier and has much more impact with the customer. The Product Demonstration at the end of a Sprint is where the program increment is showcased.

**Quality Assurance.** The project management definition of Quality Assurance is defined as "the preventative steps taken to increase the likelihood of delivering a deliverable and achieving the quality targets set". Examples of quality assurance tools and techniques include:

- Observation of project processes
- Product Review checklists



- Referencing historical data to understand areas where quality issues are likely to occur
- Reiterating the quality standards to be met to clarify the level of quality required
- Recruiting skilled staff to produce the deliverables and undertake the processes
- Conduct Peer Reviews and Quality Product Reviews to provide confidence in the quality of the project artifacts.
- Performing formal Change Control to minimize the likely number of quality issues

Once the features are built through epics and user stories, service teams are able to immediately execute the testing plans. At a minimum, agile teams are writing unit tests and doing continuous integration. The service teams know at every moment of the project how well the code is performing against the requirements. Quality Assurance activities include:

- Automated Testing
- Burn-Down Charts
- Iterative Testing against acceptance criteria
- Sprint Review
- Daily Stand-Up

**Quality Control.** The PMBOK definition of Quality Control is the “monitoring the specific project results to determine whether they comply with relevant quality standards and identifying ways to eliminate causes of unsatisfactory performance.” Agile provides more opportunities to monitor what is going on than traditional methods and hence offers more effective opportunities to intervene. The main technique is to track what software has been incrementally delivered. The Product Demonstration at the end of a sprint is where the product increment is show cased. With Agile, even though quality is the focus from the very beginning in an agile project, we still seek to validate outcomes and formally track the quality of the product we are building. Agile monitoring and control comes in a variety of forms:

- Frequent delivery
- Colocation
- Daily Stand-Up Meeting
- Sprint Review Meeting
- Requirements Traceability
- Automated Testing
- Releasable Software in every Sprint
- Code Reviews

**Improve Quality.** In Agile, quality improvement refers to anything that enhances an organization’s ability to meet quality requirements. It covers product improvement, process improvement and people based improvement. Organizationally, a continual

improvement process is an ongoing effort to improve products, services, or processes. Processes and products are constantly evaluated and improved in the light of their efficiency, effectiveness and flexibility.

After the actual level of quality has been established (through Quality Assurance and Control), the deliverables produced and the processes executed should be compared to the quality standards that have been established and quality improvement actions should be implemented as necessary. The level of quality achieved and the preventative or corrective actions undertaken should be communicated to the Project Manager for consideration and the project plan and schedule adjusted accordingly if applicable.

Retrospectives are the mechanism most service teams use for reflection. During a Retrospective the team looks at how well the sprint went and what they can do different. The high priority changes become User Stories to go into the Release Plan for implementation. Often this is the process to implement new processes.

## 2 Introduction

Quality Management, in a project context, is concerned with having the right processes to ensure both quality product and a quality project. Project Quality Management is one of the nine project management knowledge areas in the Project Management Body of Knowledge (PMBOK) from the Project Management Institute (2004).

The organization's ability to deliver new functionality with the quickest sustainable lead-time and to be able to react to rapidly changing business environments is dependent on the quality of the solution – both the product and the processes. The Agile Manifesto is focused on quality as well:

Continuous attention to technical excellence and good design enhances agility

### 2.1 Purpose

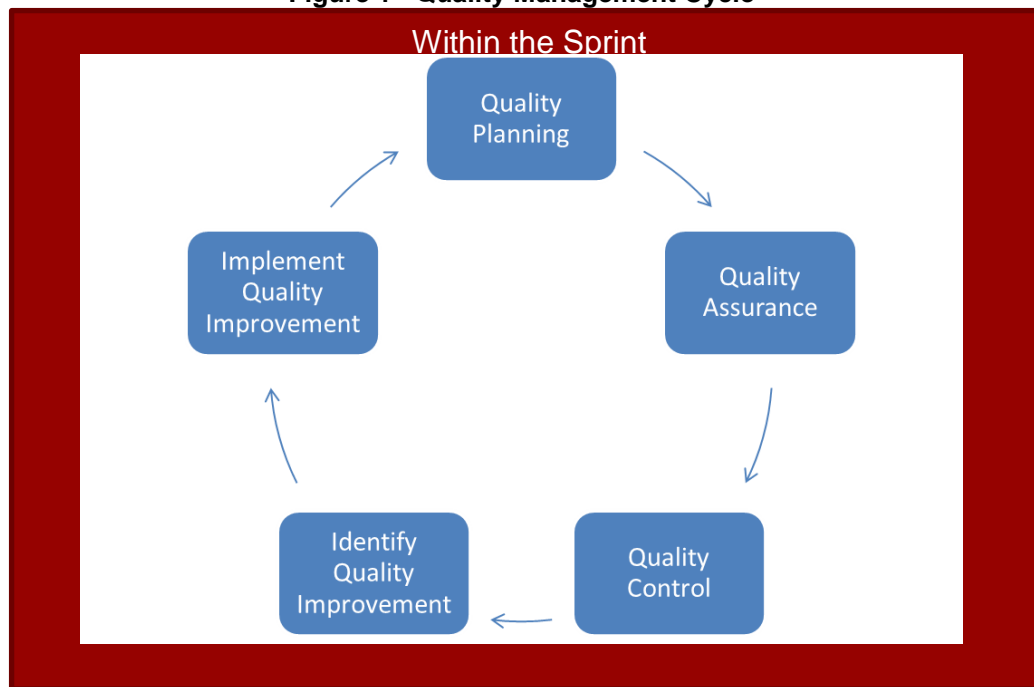
The primary purpose of the CWDS Quality Management Plan (hereafter called the "QMP") is to define how quality will be managed throughout the project lifecycle of the Child Welfare Services-New System (hereafter called "CWS-NS") Project in the following areas:

- **Quality Planning** – determining a plan for quality by documenting quality standards and framework. In Agile, quality planning tells us what we expect to do.
- **Quality Assurance** – an executing process that is necessary to continuously improve activities and processes to achieve quality. In Agile, quality assurance means ensuring that software is working right. It includes planning activities to demonstrate quality.
- **Quality Control** – a monitoring and control process that ensures every deliverable and work product is measured and tested to ensure results conform to quality standards. In Agile, quality control focuses on monitoring progress through velocity and tracking what software has been incrementally delivered. It involves implementing the plans made during quality assurance activities.
- **Quality Improvement** – adjusting a process for continuous corrective action or process improvement. In Agile, quality improvement is accomplished through sprint retrospectives.

The Agile process itself provides quality assurance and control. Agile builds quality in to the product through a combination of practices from Project Executions through Monitoring and Control.

Figure 1 below describes the Quality Management Cycle, starting with quality planning and cycling through continuously to quality improvement.

Figure 1 - Quality Management Cycle



Quality Management is focused on product and service quality and the means to achieve it. The QMP defines how the project will assess quality in its products and work processes. Quality management activities ensure that:

- Products are built to meet agreed-upon standards and requirements and meet a fitness-for-use standard.
- Work processes are performed efficiently and as documented.
- Non-conformances found are identified and appropriate corrective action is taken.

**Quality Management** applies to project products and project work processes. **Quality Control** activities monitor and verify that project deliverables meet defined quality standards. **Quality Assurance** activities monitor and verify that the processes used to manage and create the deliverables are followed and are effective. **Quality Improvement** activities seek to ensure that there is continuous improvement of quality processes and procedures and an ability to respond to corrective actions resulting from audits and reviews.

## 2.2 Scope

The QMP achieves the following objectives:

- Defines the goals of the CWS-NS Quality program and processes that will be subject to quality control.
- Identifies the activities and processes used to manage quality.

- Defines the quality management methodologies, best practices, roles and responsibilities, training and communication required throughout the life cycle of the CWS-NS project.
- Ensures all project products and artifacts conform to this plan.
- Defines the quality planning, quality assurance, quality control and quality improvement processes.

### 2.2.1 In-Scope

Quality Management processes and deliverables will be managed throughout the project lifecycle. This document defines the roles and responsibilities, standards, methods, and reporting requirements that shall be used on the CWS-NS Project.

Quality is an iterative process that consists of the following Quality Management Activities appears below:

**Table 1 - Quality Management Activities**

Quality Activity	Purpose	Frequency	Output
Peer Review	A Quality Assurance Internal Review of documents by the owner organization against defined standards.	As per the Project Schedule	Comment Log Matrix as defined in the CWS-Document Management Plan
Sprint Review	A Quality Assurance examination of service team products to verify compliance to standards.	At the end of each Sprint	Sprint Review Presentation or Demonstration
Process Checklists	A Quality Planning Activity where the Quality Manager documents the specific criteria used to evaluate each project management process	As per the Project Schedule	Quality Process Audit Checklist
Quality Process Audit	A Quality Control examination of project management processes, high-level development processes, and day-to-day practices to verify compliance to project standards.	As needed	Quality Process Audit Results
Data Analytics Assessment	A Quality Control periodic review and presentation of metrics	As needed	Metrics Results

Quality Activity	Purpose	Frequency	Output
	measurements as prescribed in the data analytics of each service team in Pivotal tracker.		
Sprint Retrospectives	A Quality Continuous Improvement activity that reviews what went well and what can be improved for each sprint. Issues are identified and the team determines appropriate corrective action or process improvement.	At the end of each Sprint	Sprint Retrospective Results  Could result in a new story in a future sprint
CWDS Digital Standards	The CWDS Digital Service Standard establishes the criteria that the CWDS digital services must meet to ensure our services are simple, fast and easy to use. Meeting the criteria means we can consistently provide high quality services and satisfy our users' needs. The Digital Service Standard applies to all CWDS digital services teams.	Used by the development teams during sprint design and development	CWDS Digital Services Standard
CWDS Digital Services Playbook	Builds upon the Digital Services Standard and defines for each standard the description and use, checklist of activities that must be executed and key questions to assess compliance.	Used by the development teams during sprint design and development	CWDS Digital Services Playbook

### 2.2.2 Out of Scope

The QMP does not include detailed process steps and procedures for Solution Quality, as this is primarily the responsibility of each digital service team's development vendor. Solution Quality is managed by the CWS-NS project by monitoring compliance with coding or engineering standards, by conducting acceptance testing to verify if the potentially shippable increment (PSI) meets the State's documented digital service standards, and with inspection test quality control activities. Each digital service development team is responsible for ensuring that quality engineering practices are used

throughout the life of the digital service and that quality is built into the service rather than being tested in at a later date.

## 2.3 Assumptions and Constraints

Table 2 – Assumptions and Constraints related to Quality

Assumptions	Constraints
<ul style="list-style-type: none"> <li>Quality Management will follow PMBOK and CA-PMF framework, along with representative IEEE standards.</li> </ul>	<ul style="list-style-type: none"> <li>Development teams are responsible for the delivery of a Quality Management solution that manages their internal quality processes and procedures and defines solution quality, which must integrate with the State’s quality management processes.</li> </ul>
<ul style="list-style-type: none"> <li>Where possible, the QMP will highlight the difference between the waterfall approach to quality processes and the Agile approach.</li> </ul>	<ul style="list-style-type: none"> <li>Quality Management will not follow ISO 9001:2008 International Standard for Quality Management based on the maturity of the organization and the ability to implement the standard.</li> </ul>

## 2.4 Integration with Quality Management Procedures

Procedures that are referenced in this Plan:

- [Quality Management 100 Quality Checklist Procedure](#)
- [Quality Management 200 Quality Product Review Procedure](#)
- [Quality Management 300 Quality Process Audit Procedure](#)

## 2.5 Document Maintenance

The CWDS Quality Management Plan will be updated as processes and procedures change. A minor version change does not change the intent of the document and consists of spelling, grammatical and minor corrections. A major version is when a document’s content is changed and represents a change in intent, change in process, or procedures. Please refer to the CWDS Configuration Management for further detail on version control.

## 2.6 Transition to Agile

The initial baselined version of the Quality Management Plan was written with an SDLC waterfall approach to quality management. With the pivot to Agile in December 2015, the quality management approach needed to be aligned to fit within the new governance

structure (roles and responsibilities) as well as adopt a more collaborative approach to quality management at the service team level.

In Agile, we define Product Quality as “fitness for use” as opposed to the more traditional approach of defining Product Quality as “conformance to requirements”. Agilists want to “satisfy the customer” and deliver “valuable software”. Traditionalists want to deliver software that implements the contracted requirements specification. Agile is very much concerned about product quality in the sense of “Fitness for use” rather than “conformance to requirements”. The first principle behind the Agile Manifesto is:

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

This principle manages to include both satisfied customers and a valuable product, both very quality oriented aspirations. In addition, the other element of this principle, continuous delivery, is the key feature of Agile and this principle is very clearly tying continuous delivery to customer satisfaction.

Project quality includes “things like applying proper project management practices to cost, time, resources, communication etc. It covers managing changes within the project”. Agile Project Quality is addressed through the traditional project management processes:

- Project Planning
- Project Estimating
- Project Execution
- Project Monitoring and Control
- Quality Management
- Risk Management
- Change Management

The struggle with an alignment to agile from a traditional waterfall approach as it applies to quality management and quality assurance:

1. Scope Creep – Requirement changes and updates are inherent to Agile methodology
2. Inadequate time to prepare test plans
3. Minimal requirements documentation to prepare the test cases
4. Highly compressed test execution cycles
5. Minimal time for regression testing
6. Change of role from being a Gatekeeper of Quality to being a Partner in Quality
7. Less emphasis on traditional project documentation (Quality Plan, Test Plan, etc.)



## 2.7 Quality Policy

The Quality Policy for the CWS-NS Project is “the CWS-NS Project shall adopt processes and practices that yield consistent and expected results when measured in terms of customer expectations, cost, schedule, and performance.”

## 2.8 Goals and Objectives of the Quality Program

In order to achieve quality in CWS-NS processes and products, quality must be built into each process and product as they are being developed and all CWS-NS team members are responsible for the quality of their output.

The CWS-NS Quality Program establishes the quality strategy to implement a system of performance indicators/metrics that communicates the performance of different aspects of the project, and defines a method of monitoring those indicators to determine and report performance trends.

The main components of the CWDS quality program are listed below:

### 2.8.1 Common Vision

The CWS-NS Project will manage both the project management and the system development activities iteratively, or in an agile manner. As such, there are some specific Agile values that must be aligned with the CWD-NS project to ensure success:

***Individuals and interactions over processes and tools***  
***Working software over comprehensive documentation***  
***Customer collaboration over contract negotiation***  
***Responding to change over following a plan***

*That is, while there is value in the items on  
the right, we value the items on the left more.*

### 2.8.2 Measurement Criteria

Measuring project health on an Agile project is very different than measuring health of a waterfall project. In waterfall, the primary criteria for determining project health and progress is Scope, Schedule and Cost. Not true with agile projects.

On CWS-NS, we must define the specific assessment areas that if measured, will provide a good representation of the basic health and progress of the project. When deficiencies of issues are identified, there must be a clear path to continuous improvement, either in the form of a corrective action or a process improvement. To that end, QA has defined the following assessment areas that will be evaluated regularly to determine overall project health and ensure that quality is being built into the product in every step. These

assessment areas are in alignment with the Checks and Balances Oversight Team processes.

**Table 3 - Measuring Quality in each Assessment Category**

Process Area	Description of Assessment Measurement
Release Integrity	<p>Includes an assessment of Release and the Program Increment planning and execution. The Product Backlog in the next PI or release is evaluated to ensure that scope and goals are defined and understood.</p> <p>The effectiveness of the release planning process and communication of release goals is reviewed in the areas of Planning, Execution and Quality. MS Project Schedule and Pivotal Tracker/JIRA are evaluated for alignment with major milestones.</p>
Engineering Practices	<p>Includes an assessment of basic engineering practices in alignment of best practices and standards, including environments, static code analysis, automated unit/system/acceptance test coverage, coding standards, environments, and continuous integration/build.</p>
Agile Processes	<p>Includes an assessment of project dynamics, adherence to scrum rules and practices, and understanding of the project goals and objectives, and working tested code that meets functional requirements at the end of each sprint.</p>
Governance	<p>Includes an assessment of whether decision making is occurring at the right level of the organization, if service managers and their teams feel empowered to address risks and issues, and if teams understand what needs to be escalated and to whom.</p>
People and Teams	<p>Includes the tracking of State resources by fiscal year and vendor resources by month (planned versus actual), as well as calculates the vacancy rate and turnover rate. Assess satisfaction with vendor teams and if the right skill sets are responsible for the right efforts.</p>
Go Live Readiness	<p>Includes an assessment of implementation readiness for the next release of functionality to the users, specifically training, county readiness, go live readiness and organizational change management.</p>
Customer Value	<p>Includes an assessment of surveying internal and external stakeholders to gauge their satisfaction with the level of communication that is provided regarding the upcoming release, the value of the functionality that is planned and the perceived quality of the product.</p>
Technical Debt Management	<p>Includes an assessment of the accumulation of issues with the code or the hardware, escapee tracking, tracking the quality of the</p>

Process Area	Description of Assessment Measurement
	product and code and tracking the integration into third party or external systems.
Information Exchange	Includes data quality, Interface and partner management, legacy related risks and Data Conversion.
Project Management Metrics	<p><b>Risk and Issue Management</b></p> <ul style="list-style-type: none"> <li>• Trending Functional categories of issues</li> <li>• Average age of open issues</li> <li>• Trending Functional categories of risks</li> <li>• Average age of open risks</li> <li>• Time to define mitigation steps</li> </ul> <p><b>Change Management</b></p> <ul style="list-style-type: none"> <li>• Number of open and closed change requests for the reporting period (by service team).</li> <li>• Average age of change requests</li> <li>• Change Control process timeline from submission through implementation</li> <li>• Adherence to change notification versus change request process</li> </ul> <p><b>Schedule Management</b></p> <ul style="list-style-type: none"> <li>• Target % Complete versus % Complete of Milestone tasks</li> <li>• Average Sprint Velocity</li> <li>• Target % Complete versus % Complete of each Program Increment or release (by either the Potentially Shippable Product [PSI] or Minimum Viable Product [MVP]).</li> </ul> <p><b>Deliverables Management</b></p> <ul style="list-style-type: none"> <li>• Contractual deliverable Timeliness</li> <li>• Deliverable Acceptance Rate</li> </ul> <p><b>Quality Management</b></p> <ul style="list-style-type: none"> <li>• Percentage of compliant Quality Process Audits</li> <li>• Percentage of compliant Quality Product Reviews</li> <li>• Percentage of compliant Quality Inspections</li> </ul>
Technical Metrics	<p><b>Velocity</b></p> <ul style="list-style-type: none"> <li>• Is velocity increasing, decreasing or stabilizing?</li> <li>• Average velocity over the last three sprints</li> </ul> <p><b>Volatility</b></p> <ul style="list-style-type: none"> <li>• The rate of change in velocity over time</li> </ul> <p><b>Burn-Down</b></p> <ul style="list-style-type: none"> <li>• Percentage of time the development team finishes early sprint after sprint because they aren't committing to enough work.</li> <li>• Percentage of time the team misses their forecast sprint after sprint because they're committing to too much work.</li> <li>• Trending of steep drops rather than a more gradual burndown because the work hasn't been broken down into granular pieces.</li> <li>• Percentage of time the product owner adds or changes the scope mid-sprint.</li> </ul> <p><b>Design</b></p>

Process Area	Description of Assessment Measurement
	<ul style="list-style-type: none"> <li>• Percentage of bugs identified that are not in compliance to coding standards or guidelines</li> </ul> <p>Test</p> <ul style="list-style-type: none"> <li>• Trending of bugs in each sprint</li> <li>• Percentage of recurring bugs not fixed the first time (reintroduced)</li> <li>• Percentage of production incidents</li> </ul>

### 2.8.2.1 Quality Checklists

In alignment with the above assessment categories, QA will develop a checklist for each project management and technical plan or process in accordance with [Quality Management Procedure 100: Quality Checklists](#).

### 2.8.2.2 Quality Product Review

In alignment with the above assessment categories, QA will review the appropriate deliverables, work products and documentation in accordance with [Quality Management Procedure 102: Product Reviews](#).





### 2.8.2.3 Quality Process Audit

In alignment with the above assessment categories, QA will assess the execution of project management and technical processes in accordance with [Quality Management Procedure 101: Process Audit](#).

## 2.8.3 Assessment Category Stoplight Status Legend

Quality Measurements on reports or dashboards may be rated per a Green, Yellow or Red stoplight status. Each stoplight status has specific criteria to gauge the measurement, listed below.

**Table 4 – Assessment Category Stoplight Status Legend**

Legend			
 <p>Satisfactory – No corrective action necessary.</p> <p>Risks/Issues may exist and contain appropriate mitigation or resolution steps and the project is addressing them.</p> <p>No impact to project schedule, budget or quality.</p> <p>The current risk to the overall project and quality is low.</p>	 <p>Caution – There may be a need for corrective action now or quite soon.</p> <p>Risks/Issues often are not identified or contain stale mitigation or resolution steps.</p> <p>Moderate impact to schedule, budget or quality.</p> <p>The current risk to the overall project and quality is Moderate.</p>	 <p>Critical - Escalated for immediate corrective action.</p> <p>Risk mitigation is not effective or issue resolution is not timely. Critical processes are not effective and are resulting in delays or rework.</p> <p>Significant impact of 10% or more to schedule or budget.</p> <p>There is a significant risk to the project.</p>	 <p>Not enough data is available to decide about project health.</p>

## 2.8.4 CWS-NS Products and Services

Digital Service Development teams are responsible for defining the Sprint-level definition of Done (DoD), the Release-level definition of Done (DoD) and user story acceptance criteria.

However, the Sprint-level Definition of Done or the Release-level Definition of Done only tells half the story. It's about building the right product, at the right time, for the right market. Staying on track throughout the program means collecting and analyzing meaningful data along the way. In any agile program, it's important to track both business metrics and agile metrics. Business metrics focus on whether the solution is meeting the market need, and agile metrics measure aspects of the development process.

The CWS-NS quality program performs assessments to determine adherence to standards, best practices and the Agile Manifesto for all products and services. The data is then analyzed and evaluated by quality assurance to determine corrective action or remediation activities to bring the performance back in line with expectations for the product or service.

All CWDS products must comply with the CWDS Digital Services Standards and the Digital Services Playbook.

## 2.8.5 Establishing Quality Processes

### 2.8.5.1 Project Management Documentation

For project management processes that have been previously defined and documented in approved plans, the Quality Assurance Manager will review and assess the plan using the procedures described in the **Quality Management Procedure 200: Quality Product Review**. For those project management processes that have not been previously defined, the Quality Assurance Manager will assist in establishing effective quality processes by providing support and guidance to the CWS-NS Project Team. These developed processes will be in compliance with industry standards and best practices. QA activities in this area will include:

- Provide support and guidance to CWS-NS Project organizations, in developing project quality processes, standards, templates, etc., that comply with appropriate industry standards and best practices.
- Participate in CWS-NS Process Definition and Peer Review activities as appropriate; to ensure all CWS-NS processes comply with project and industry standards and best practices.
- Conduct Product Reviews of all internal plans and processes and provide meaningful feedback to the CWS-NS team.

### 2.8.5.2 Service Team Artifacts

For all development team artifacts, the Quality Assurance Team will review and assess the plan or process using the procedure described in the **Quality Management Procedure 200: Quality Product Review**. Once the artifact has been approved by DCA

and where applicable, the Quality Assurance Team will also review and assess the execution of the plan or process using the procedure described in the **Quality Management Procedure 300: Quality Process Audit**. QA activities in this area will include:

- Provide support and guidance to Development Team in developing project quality processes, standards, templates, etc., that comply with appropriate industry standards and best practices.
- Participate in Development Work Order Authorization review, process definition and product review activities as appropriate; to ensure all processes comply with project and industry standards and best practices.
- Conduct Product Reviews of all development team artifacts and provide meaningful feedback to the team.

### 2.8.6 Establishing Quality Improvement

The CWS-NS quality program will provide assistance in development of a CWS-NS process improvement program. Repeatable processes and procedures will be assessed to determine their effectiveness. Products will be evaluated to determine corrective action. Recommendations for process improvement or corrective action will be provided. To aid in the establishment and direction of a process improvement program, the quality program will:

- Work with the CWS-NS Project Team personnel to establish an effective and comprehensive Process Improvement Program.
- Verify that the Process Improvement is consistent with project plans, procedures and standards.
- On a periodic basis collect and analyze process measurement data collected from the project.
- Provide recommendations, based on the analysis of process measurement data, for process improvement actions.
- Monitor the implementation of process changes, and modify process measures as necessary.

### 3 Roles and Responsibilities

The following tables describe the roles and responsibilities of the CWS-NS Project stakeholders in the Quality Management arena.

**Table 5 - Quality Management Stakeholder Matrix**

STAKEHOLDER ROLE	RESPONSIBILITY
CWS-NS Project Team	<ul style="list-style-type: none"> <li>- Identify, report, review and/or analyze project deliverables and/or work products, focusing on quality characteristics such as completeness, consistency, fitness of use, etc.</li> </ul>
Technology Manager	<ul style="list-style-type: none"> <li>- Establish and execute technical policies, processes, and procedures and ensure adherence to defined quality standards.</li> <li>- Communicate project status, quality issues and risks to the project management team.</li> <li>- Maintain the CWDS Digital Service Standards and the CWDS Digital Service Playbook.</li> </ul>
Project Manager	<ul style="list-style-type: none"> <li>- Establish and execute project management policies, processes, and procedures and ensure adherence to defined quality standards.</li> <li>- Communicate project status, quality issues and risks to the project management team.</li> <li>- Proactively seek continuous improvement opportunities for corrective action or process improvement.</li> </ul>
Checks & Balances Team	<ul style="list-style-type: none"> <li>- Provide independent oversight of quality issues and areas of non-conformance in accordance to CA-PMF and IEEE.</li> <li>- Participate as needed in quality audits and quality reviews</li> <li>- Identify quality risks or issues</li> </ul>
Project Director	<ul style="list-style-type: none"> <li>- Communicate quality issues and risks to internal and external stakeholders.</li> <li>- Communicate with project staff regularly to direct project activities and stay abreast of project quality status.</li> <li>- Communicate with ELT to report any quality related issues.</li> <li>- Participate in the establishment and oversight of the project's quality management effort.</li> </ul>
Service Manager or Product Owner	<ul style="list-style-type: none"> <li>- Responsible for the overall quality usage of the service team products and services.</li> <li>- Works with the Development Team to determine the Definition of Done (sprint and release).</li> <li>- Approves user story acceptance criteria</li> <li>- Accepts or Rejects user stories in the sprint backlog based on quality.</li> <li>- Participate in service team sprint reviews to identify quality issues or concerns.</li> <li>- Escalates quality issues or risks to the Project Director as needed.</li> </ul>
Scrum Master	<ul style="list-style-type: none"> <li>- Communicate quality issues and risks to internal and external stakeholders.</li> <li>- Communicate with project staff regularly to direct project activities and stay abreast of project quality status.</li> <li>- Monitor milestones, activities, timelines, resources, budgets and critical path</li> <li>- Develop and track project metrics and/or data analytics</li> </ul>
CWDS Quality Assurance	<ul style="list-style-type: none"> <li>- Identify and escalate any critical project issues to the Project Manager.</li> <li>- Identify Quality Standards and Metrics for PM Processes.</li> </ul>

STAKEHOLDER ROLE	RESPONSIBILITY
	<ul style="list-style-type: none"> <li>- Provide QA inputs for developing project work products and ensuring that quality targets are defined for each deliverable and process.</li> <li>- Evaluate project processes and procedures and provide evaluation reports related to standards compliance, process variances, and identify process improvement opportunities.</li> <li>- Participate in service team sprint reviews to identify quality issues or concerns.</li> <li>- Implement Quality Assurance techniques to ensure the quality of the outputs produced by the project</li> <li>- Implement Quality Control techniques to control the quality of the outputs produced by the project</li> <li>- Identify quality deviations and improvement actions for implementation</li> <li>- Audit Project Office processes and artifacts.</li> <li>- Maintain the Quality Management Plan</li> <li>- Collect and analyze project metrics.</li> <li>- Provide written reports related to standards compliance, identify of process improvement opportunities, correctness, completeness, anomalies and recommendations.</li> <li>- Establish and maintain a repository for quality measurement and tracking</li> <li>- Escalates quality issues or risks to the Service Manager as needed.</li> </ul>
Digital Service Quality Engineers	<ul style="list-style-type: none"> <li>- Ensure compliance to the Digital Standard and the Digital Playbook.</li> <li>- For each service team, oversee the development team Quality Management Program to ensure quality objectives for the new system are satisfied, and pass quality reviews.</li> <li>- Establish coding and engineering standards.</li> <li>- Conduct coding, engineering and testing inspections as needed.</li> <li>- Support Requirements Traceability Planning.</li> <li>- Participate in service team sprint reviews to identify quality issues or concerns.</li> <li>- Escalates quality issues or risks to the Product Manager as needed.</li> <li>- Establish reporting standards that provide findings from quality measurements on a periodic basis identifying areas where business, technical, and/or management quality objectives are or are not being met, or where trends in quality are moving in or out of control limits.</li> </ul>
Digital Service Development Teams	<ul style="list-style-type: none"> <li>- Work with the Quality Engineers to integrate quality processes into the CWS-NS quality program.</li> <li>- Participate in quality reviews and audits.</li> <li>- Respond to quality review and audit findings as part of the quality improvement process.</li> <li>- Inform the project's technical lead and quality engineers of any quality issues, quality management, system testing, system change requests, problem reporting and project requirements and definition.</li> <li>- Communicate with project's technical staff and project technical consultants on quality related issues.</li> <li>- Define testing acceptance criteria for performing unit, system and performance testing when products are delivered to the pre-production environment as candidates for release to production.</li> </ul>
CWDS Release Manager	<ul style="list-style-type: none"> <li>- Define testing acceptance criteria for performing acceptance testing when products are delivered to the pre-production environment as candidates for release to production.</li> <li>- Facilitate user acceptance validation on pre-production releases.</li> </ul>



STAKEHOLDER ROLE	RESPONSIBILITY
	<ul style="list-style-type: none"> <li>- Ensure that detailed testing is performed on all technical areas such as system and data interfaces, integration with external systems, data interfaces and automated workflow.</li> </ul>

## 4 Quality Management Approach

The quality management process is a method by which the quality of CWS-NS deliverables and processes is declared and controlled during the project life cycle. This process requires completing a variety of review techniques and implementing a set of corrective actions to address any deficiencies and raise the quality levels within the project.

The Quality Management Process involves:

- Identifying the types of quality measurement techniques to be undertaken in quality product reviews and quality process audits.
- Measuring deliverable and process quality (via Quality Assurance and Quality Control).
- Taking action to enhance the level of deliverable and process quality.
- Reporting the level of quality attained to project management.

Although Quality Assurance methods will be initiated during Sprint Planning and Execution, Quality Control techniques are implemented during Sprint Execution during design and development activities. Without a formal Quality Management Process in place, the basic premise of delivering the project to meet 'time, cost and quality' targets may be compromised.

Establishing the quality management program for CWS-NS involves five general steps, defined in further detail below:

1. Determine approach and strategy for Agile Quality Management
2. Plan Quality (Quality Planning)
3. Measure and Assess Quality
4. Quality Assurance
5. Control Quality (Quality Control)
6. Improve Quality (Quality Continuous Improvement)

### 4.1 Determine Approach for Agile Quality Management

As stated before, Quality Management is a method for ensuring that all the activities necessary to design, develop and implement a product or service are effective and efficient with respect to the system and its performance. Quality management can be considered to have three main components: quality control, quality assurance and quality improvement. Quality management is focused not only on product quality, but also the means to achieve it.

The approach for quality management needs to include two components: project quality and product quality.

### 4.1.1 Project Quality

**Project Quality** is the “things like applying proper project management practices to cost, time, resources, communication etc. It covers managing changes within the project”.

### 4.1.2 Product Quality

**Product Quality** is making sure that the product or service delivered is ‘fit for purpose’ and covers things like how well it meets the user’s needs, and the total cost of ownership. Fitness for use can only be defined by the customer. Agile is very much concerned about product quality in the sense of “Fitness for use” rather than “conformance to requirements”. One of the principles behind the Agile Manifesto is:

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

This principle manages to include both satisfied customers and a valuable product, both very quality oriented aspirations. The other element of this principle, continuous delivery, is the key feature of Agile and this principle is very clearly tying continuous delivery to customer satisfaction.

## 4.2 Plan Quality

Quality Planning involves identifying which organizational standards and which industry standards are relevant to CWS-NS and how to satisfy them. The process outlines the rules that define the quality needs of the project, the required standard for the project’s deliverables and processes and how it will be confirmed that the planned requirements are provided in the project’s final product.

PMBOK® defines Quality Planning as *“the process of identifying quality requirements and/or standards for the project and product, and documenting how the project will demonstrate compliance. Quality planning should be performed in parallel with the other project planning processes. For example, proposed changes in the product to meet identified quality standards may require cost or schedule adjustments and a detailed risk analysis of the impact to plans.”*

Agile Project Planning tells us what we expect to do and is usually accomplished in the Sprint Planning activity and throughout the sprint in Backlog Refinement. The job of the Agile Project Manager/Scrum Master is to guide the service team to successful delivery despite the challenges the world throws at the project.

Quality Planning is really about the initial set of assumptions that we make as an agile team about how we are going to manage quality – on our project, in our documentation, in the delivery of our product. As it relates to developing software, quality planning has mostly been done for us. Because we are agile, quality is implicit... it is understood by virtue of the fact that we are using an agile methodology.

When we have discussions about doing test driven development, pair programming, or continuous integration; we are making decisions about how we are going to handle quality. The decision to make use of acceptance criteria is simply a decision on how we

will know we have met the requirements of our stakeholders. Service teams work with their development vendor to ask the following questions:

- Are we going to do unit testing?
- How about manual regression?
- Will we need to test for performance, scalability, or security?
- How will we know we have met any applicable policy or legislative requirements?

As service teams have these conversations, they are implicitly planning quality into their sprints and delivered products. As quality is planned for each service team, the following components are considered:

### 4.2.1 Establishing Quality Standards

We establish quality standards through the following methods:

- California Project Management Framework (CA-PMF) for project management processes.
- Approved approach and methodology for System Development (design and development standards) as stated by the Digital Service Development Team.
- [CWDS Digital Services Standard](#) and [CWDS Digital Service Playbook](#) for design and development standards.
- IEEE standards for technical processes and procedures.

### 4.2.2 Defining Acceptance Criteria

All user stories should have acceptance criteria defined. Acceptance criteria is the confirmation that the user story has been satisfied. Acceptance criteria are short and easy to understand statements that are testable. The statements detail what the user expects the product/feature to do. Acceptance criteria enable developers to understand that they have satisfied the user requirement. Acceptance criteria provide a detailed scope of the requirement, which help the team to understand the value and help the team to slice the user story horizontally.

Acceptance Criteria, or condition of satisfaction, help to set expectations within the team as to when a team should consider something done. It helps the service team to break down the user stories in task(s), if acceptance criteria are defined in detailed, then the team can provide better effort estimation for a user story, so development cycle can be shorter, resulting in no waste.

Before the developer can mark the User Story as 'done', all criteria must be fulfilled so that it is ensured that the User Story works as planned and tested. The service manager is usually responsible for specifying what the acceptance criteria should be for each of the user stories.

### 4.2.3 Agree on the Definition of Done

The Definition of Done (DoD) is a simple list of activities (writing code, coding comments, unit testing, integration testing, release notes, design documents, etc.) that add

verifiable/demonstrable value to the product. Focusing on value-added steps allows the team to focus on what must be completed in order to build software while eliminating wasteful activities that only complicate software development efforts.

DoD is a global definition of what a team needs to complete for their product or increment to be ready for release. For a development team, this usually encompasses items like: code is checked in; code is merged; unit tests are written, executed and pass.

Scrum asks that teams deliver “potentially shippable software” at the end of every sprint. To me, potentially shippable software is a feature(s) that can be released, with limited notice, to end users at the product owner’s discretion. Products that can be released to end users with two days can be reasonably said to be in potentially shippable state. Ideally, potentially shippable is equivalent to the Definition of Done. In reality, many teams are still working towards a potentially shippable state. Such teams may have a different DoD at various levels:

- Definition of Done for a feature (story or product backlog item)
- Definition of Done for a sprint (collection of features developed within a sprint)
- Definition of Done for a release (potentially shippable state)

The DoD is a comprehensive checklist of necessary, value-added activities that assert the quality of a feature and not the functionality of that feature.

### 4.3 Measure and Assess Quality

It is often said that, "What gets measured gets done." Measurements communicate values and priorities to the CWS-NS organization. This activity refers to the process of performing quality assurance and conducting quality control activities to assess the actual level of quality of each deliverable and process undertaken within the project.

Agile provides more opportunities to monitor what is going on than traditional methods and hence offers more effective opportunities to intervene (i.e. control). Traditional project monitoring focuses on tracking how much effort is expended on each task. The main Agile monitoring technique is to track what software has been incrementally delivered. This is both easier and has much more impact with the customer. The Product Demonstration at the end of a Sprint is where the product increment is show cased.

Quality Metrics are parameters or ways to quantitatively assess a project’s level of quality, along with the processes to carry out such a measurement. Metrics outline the standard that work will be measured against and are often unique to each project. CWS-NS Quality Metrics will be defined in the planning phase of the project and then measured throughout the project’s life to track and assess the project’s level of conformity to its established quality baselines.

When identifying metrics by which to measure project quality against, an established standard is identified and then used to establish a quality baseline for each defined quality metric. This baseline is then used as a barometer to measure overall project quality throughout the project’s life. Sources of quality baseline information include the CWS-NS Quality Management Plan, similar projects completed and Industry Standards.

### 4.3.1 Effective Measurements

Agile also offers several other ways to monitor progress:

- Velocity
- Hit Rate
- Work Remaining on Tasks
- Sprint Burndown

#### 4.3.1.1 Velocity

Velocity is a measure of the service team's capacity to delivery in a sprint. The unit of measure is the same as that used for estimating Tracking Velocity is the closest to the traditional tracking of effort. But Velocity is a single number for the entire team for the entire sprint so tracking this is much simpler than tracking a multitude of discrete tasks.

The predicted Velocity for the first sprint, created during Agile Project Initiation, is guessed or based on data from previous projects. After that the actual Velocity is measured at the end of each sprint. The first step is to total the estimates for the User Stories completed in the sprint to get the actual Velocity for that sprint. This actual measure of Velocity is used to predict the Velocity for subsequent sprints in the Release Plan. Agile processes promote a sustainable pace, so the Product Owner can't cram more into a sprint than the Velocity allows.

Task level Velocity works in a similar way during the Sprint Planning Meeting. It is used to validate the amount of work the Agile Team takes on for the sprint. It will become apparent after only a couple of sprints if the Agile Team has sufficient capacity to deliver the essential functionality by the deadline. If it looks like the team won't be able to deliver then corrective action can be taken, which usually means cutting scope.

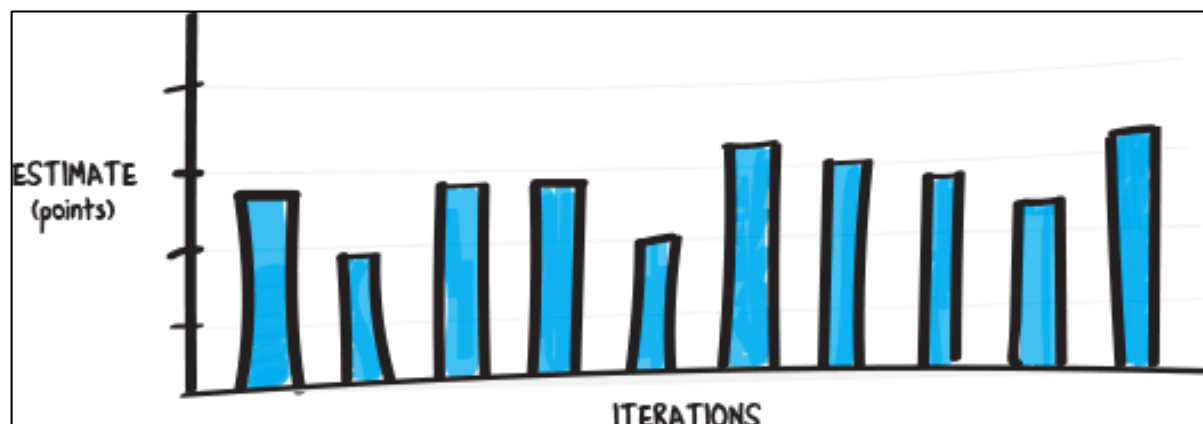


Figure 2 - Velocity

It's important to monitor how velocity evolves over time. New service teams can expect to see an increase in velocity as the service team optimizes relationships and the work process. Existing service teams can track their velocity to ensure consistent performance over time, and can confirm that a particular process change made improvements or not. A decrease in average velocity is usually a sign that some part of the service team's development process has become inefficient and should be brought up at the next retrospective. When velocity is erratic over a long period of time, always revisit the team's estimation practices. Look for these patterns:

- Are there unforeseen development challenges we didn't account for when estimating this work?
- How can we better break down work to uncover some of these challenges?
- Is there outside business pressure pushing the team beyond its limits? Is adherence to development best practices suffering as a result?
- As a team, are we overzealous in forecasting for the sprint?

Since each service team's estimation culture is unique, their velocity will be as well. Quality assurance activities should resist the temptation to compare velocity across service teams. Measure the level of effort and output of work based on each service team's unique interpretation of story points.

#### **4.3.1.2 Hit Rate**

The Hit Rate is the percentage of work allocated to a sprint that was actually completed. Ideally a team should have a hit rate near 100%. A low Hit Rate indicates the team is struggling to meet their commitments; this might be caused by over commitment, impediments, etc. For example, if a team aimed for 100 Story Points of User Stories, but ended up only completing 55 Story Points, their hit rate is 55%. This might be ok for one sprint, but if the pattern repeats time and again, the team is over committing and should adjust their workload down.

#### **4.3.1.3 Work Remaining: Sprint Burn Down**

Each day the team estimates the work remaining on each active Task in their daily Stand-Up Meeting. The estimate of work remaining Task will hopefully go down over time, but it may rise if the Task is more complicated than anticipates. This is where the data for the Sprint Burn Down Chart come from.

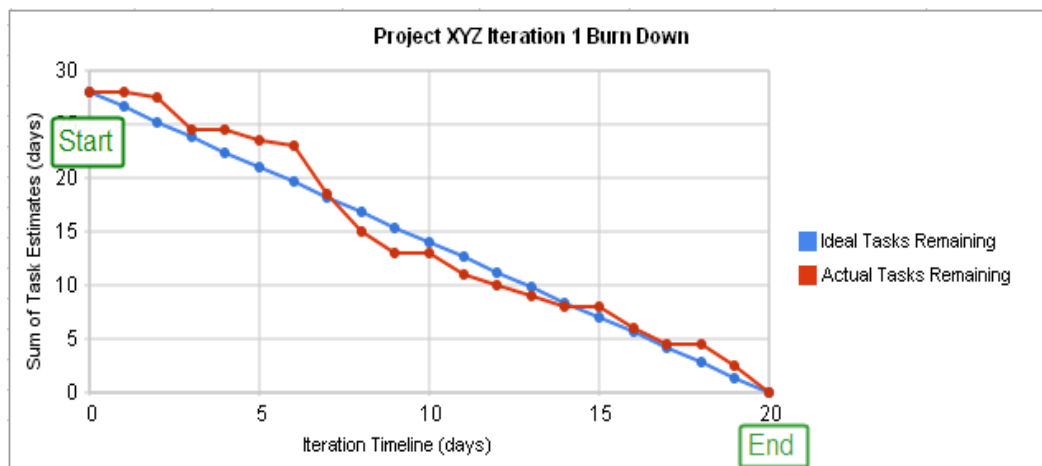
A Sprint Burn Down Chart is a method of visualizing whether the team is on track to complete the work allocated to the sprint. If the downward trench indicates not all of the work will be completed in time we have an opportunity to intervene and do something about it.

Scrum teams organize development into time-boxed sprints. At the outset of the sprint, the team forecasts how much work they can complete during a sprint. A sprint burndown report then tracks the completion of work throughout the sprint. The x-axis represents time, and the y-axis refers to the amount of work left to complete, measured in either story

points or hours. The goal is to have all the forecasted work completed by the end of the sprint. Look for these patterns:

- The team finishes early sprint after sprint because they aren't committing to enough work.
- The team misses their forecast sprint after sprint because they're committing to too much work.
- The burndown line makes steep drops rather than a more gradual burndown because the work hasn't been broken down into granular pieces.
- The product owner adds or changes the scope mid-sprint.

**Figure 3 – Sprint Burndown Chart**



#### 4.3.1.4 Work Remaining: Epic and Release Burndown

A Release Burn Down Chart does the same thing but for an entire Release. It shows if the team is on track to complete the User Stories required for the Release. Epic and release burndown charts track the progress of development over a larger body of work than the sprint burndown, and guide development for service teams. Since a sprint may contain work from several epics and versions, it's important to track both the progress of individual sprints as well as epics and versions.

"Scope creep" is the injection of more requirements into a previously-defined project. For example, if the team is delivering a new website for the company, scope creep would be asking for new features after the initial requirements had been sketched out. While tolerating scope creep during a sprint is bad practice, scope change within epics and versions is a natural consequence of agile development. As the team moves through the project, the product owner may decide to take on or remove work based on what they're learning. The epic and release burn down charts keep everyone aware of the ebb and flow of work inside the epic and version. Look for these patterns:

- Epic or release forecasts aren't updated as the team churns through the work.
- No progress is made over a period of several iterations.



- Chronic scope creep occurs, which may be a sign that the service manager doesn't fully understand the problem that body of work is trying to solve.
- Scope grows faster than the service team can absorb it.
- The team isn't shipping incremental releases throughout the development of an epic.

### 4.3.2 Measuring the Quality of Internal Project Documentation

As stated in the CWS-NS Document Management Plan, all CWS-NS internal project documents (PM Plans, processes and procedures) must conform to the OSI Best Practices standard, which is closely aligned with the PMBOK and CA-PMM framework standards.

Specific acceptance criteria will be developed for each internal Quality Product Review and the documentation must be in compliance against State standards, project standards and best practices. Some general acceptance criteria for internal project documentation includes:

- Documentation complies with OSI Best Practices and standards (which indirectly ensures alignment with PMBOK and CA-PMF)
- Documentation was reviewed by all assigned SMEs and Quality Assurance Manager for:
  - Conformance to standards
  - Internal Consistency
  - External Consistency
  - Material Deficiencies
  - Completeness
  - Fitness of Use
  - Requirements Traceability (where applicable)
- Documentation has no open major deficiencies
- Relevant documentation was updated and posted to project repository

Internal and external process documentation, as well as observation of the process and procedure execution, will have acceptance criteria defined in both the governing Plan and in the checklists that will be developed for each process. See [Quality Management 100 Quality Checklist Procedure](#) for more information.

### 4.3.3 Measuring the Quality of Development Team Deliverables

Deliverables (sprint status reports, technical documentation, and training material) that are developed by the digital service team developers as sprint deliverables or

augmented system documentation must confirm to both IEEE standards as well as the contractual terms in the vendor Statement of Work (SOW).

Specific acceptance criteria will be developed within each sprint user story for each development team deliverable and it must be in compliance against State standards, project standards and best practices. Some general acceptance criteria for internal project documentation includes:

- Deliverable complies with IEEE standards and best practices
- Plan was reviewed by all assigned SMEs and Quality Solution Manager for:
  - Conformance to standards
  - Internal Consistency
  - External Consistency
  - Material Deficiencies
  - Completeness
  - Fitness of Use
  - Requirements Traceability (where applicable)
- Plan has no open major deficiencies
- Relevant documentation was updated and posted to public repository

#### 4.3.4 Measuring the Quality of Software Code

There are many definitions of code quality, but usually people refer to "quality code" as code that is flexible, testable and readable. If we can assess how good our code is, we can estimate the effort for the next tasks more precisely or allocate time for reducing the technical debt in order to improve code quality. So by measuring the internal code quality we improve its external quality and as a result improve the software product.

Depending from circumstances, you can use different techniques to evaluate the quality of a software product:

- **Completeness:** Which part of the needed features are actually implemented.
- **Asking users:** What is the feeling of typical users about the software?
- **Metrics:** Some metrics can give you a good idea about the quality of the code (i.e. bugs per line of code, code coverage, function points, number of interfaces, number of lines of code, transaction execution time, program load time).
- **Process:** The use (or not) of certain processes is a good hint about the quality of a development process. (i.e. bug tracking, automated tests, versioning tools).
- **Bug detection:** Bug detection rate is a good indicator of the quality of the code produced.

Some software development practices we can rely on to address quality include:

- Automated integration tests.

- Unit Tests.
- Acceptance tests for every user story
- Feature Level Tests (testing groups of stories)
- Code reviews
- Risk tracking
- Manual testing for each story, immediately after it is implemented.
- Automated performance tests.

Many of these practices have come about as corrective measures learned in a sprint retrospective that indicated that things could be improved – continuous improvement is an iterative process.

### 4.3.5 Metrics

Time and resources devoted to measurement demonstrate management commitment that the object of the measurement is important. Therefore, the selection of appropriate metrics is an essential starting point for process improvement. Quality Metrics are an objective measure of the quality of a product or process or an objective measure of the effectiveness or efficiency of a system product. Quality Metrics use common language to assess progress about quality and need to be objective in order to provide clarity to the entire stakeholder pool.

As internal CWS-NS documentation, work products and code are approved, the Quality Assurance Manager and the Quality Solution Manager will begin to collect metrics on key functional areas. These metrics will be distributed in one of several ways:

- Executive Dashboard on the CWS-NS Project Website
- Executive Dashboard presented at the weekly Service Manager Meeting
- Key Metrics presented in the monthly Checks and Balances Oversight Report
- Detailed metrics communicated through the Quality Management monthly status report (MSR).

## 4.4 Quality Assurance

The project management definition of Quality Assurance is defined as “the preventative steps taken to increase the likelihood of delivering a deliverable and achieving the quality targets set”. Quality Assurance techniques are often undertaken at a summarized level of the project by an external project resource. Examples of quality assurance tools and techniques include:

- Observation of project processes
- Product Review checklists

- Referencing historical data to understand areas where quality issues are likely to occur
- Reiterating the quality standards to be met to clarify the level of quality required
- Recruiting skilled staff to produce the deliverables and undertake the processes
- Conduct Peer Reviews and Quality Product Reviews to provide confidence in the quality of the project artifacts.
- Performing formal Change Control to minimize the likely number of quality issues

In Agile, Quality assurance is about making sure we are building the right product from the very beginning. Early in the iteration, the teams meet with our customers to define exactly what is to be built. Every role on the project has this opportunity and is encouraged to be involved. There are people looking at the requirements from every conceivable angle: system architecture, development, QA, analysis and design, and usability. The problem is explored from all perspectives, before developers set off writing code, to ensure we are building a complete product.

Once the features are built through epics and user stories, service teams are able to immediately execute the testing plans. At a minimum, agile teams are writing unit tests and doing continuous integration. The service teams know at every moment of the project how well the code is performing against the requirements.

#### **4.4.1 Quality Assurance Activities**

Quality Assurance is a set of activities for ensuring quality in the processes by which products are developed. The focus is to prevent deficiencies through planned and systematic activities in a proactive approach. Quality Assurance determines compliance to project policies and procedures with the ultimate goal of quality assurance to build quality into the product or service, rather than testing it in later. Even though quality is the focus from the very beginning, service teams still seek to validate outcomes and formally track the quality of the product they are building. Some Agile activities that are used in quality assurance are:

##### **4.4.1.1 Automated Testing**

The advantage of automated testing is that a service team knows the health of the product in real time. They are able to measure and track defects and get them resolved as soon as they are introduced into the build. Manual testing, in parallel with the automated testing, gives a more intuitive way to exercise aspects of the code that are difficult to automate.

##### **4.4.1.2 Burn-Down Charts**

Another example is a scrum master that constantly tracks burn-down at the project level to see how well the team is doing against the backlog. Within the sprint, progress is tracked to make sure that the team can deliver on their commitments. Service teams

can also track defects, defect status, and test trends – all which provides the team a way to continuously control the project quality.

#### **4.4.1.3 Iterative Testing against Acceptance Criteria**

Ideally, a service team does not wait until the end of the project to test, when we have the least amount of time to actually fix a problem, or respond to a change. We know at all times the health of the project, if the team is burning hot, if defects counts are trending up or down, how well we are resolving issues, and if those issues are becoming impediments to getting new product built.

User Stories are the high level description of the external behaviors and business rules of your software. Each User Story has acceptance criteria defined and should have at least one acceptance test planned. The acceptance tests elaborate the brief description provided by the User Story. They define the scope of the story and clarify the Service Manager's intent with concrete examples. This clarifies the Service Manager's intent, points the team in the right direction, and confirm when the intent has been met. Acceptance tests should be automated, and should become the foundation of the regression tests, validating that the customer's intent continues to be met by the software after each change to the code. In Test Driven Development the tests become the specification. Because the tests are automated there is no ambiguity, the software either passes the test or it fails.

#### **4.4.1.4 Sprint Review**

Service teams review features with their customer as they are completed. They do formal product demonstrations (sprint review) and retrospectives at the end of every sprint. These processes allow the team to control not only the quality of the emerging product, but also of the processes we are using to deliver that product. Feedback from the customer and/or stakeholders is essential to making sure that the product is evolving according to the quality standards that were agreed to at the beginning of the iteration. For more information, refer to the [Agile Process 202 Sprint Review](#).

#### **4.4.1.5 Daily Stand-Up**

The last quality assurance technique is that the team holds itself accountable by meeting in a daily standup. This allows the team to stay plugged in, assess progress, and identify impediments. In addition, the team has constant access to the product owners. This constant visibility allows the customer to fine tune the solution, as it is being built, to ensure that the product will meet market requirements.

### **4.4.2 Differences between Quality Assurance and Quality Control**

Before we start with Quality Control, it may be helpful to understand the differences between Quality Assurance and Quality Control. The following table provides an overview of the differences between the quality assurance process and the quality control process.

**Table 6 - Quality Assurance versus Quality Control**

	<b>Quality Assurance</b>	<b>Quality Control</b>
Definition	A set of activities for ensuring quality in the processes by which products are developed.	A set of activities for ensuring quality in products. The activities focus on identifying defects in the actual products produced.
Focus	<b>Proactive</b> → Aims to prevent defects with a focus on the process used to make the product. Determines compliance to project policies/procedures.	<b>Reactive</b> → Aims to identify (and correct) defects in the finished product. Measures specific project results against standards.
Goal	The goal is to improve development and test processes so that defects do not arise when the product is being developed.	The goal is to identify defects after a product is developed and before it's released.
How	Establish a good quality management system and the assessment of its adequacy. Periodic conformance audits of the operations of the system.	Finding and eliminating sources of quality problems through tools and processes so that customer's requirements are continually met.
What	Prevention of quality problems through planned and systematic activities including documentation. Corrective or preventive action as a result of the audit.	The activities or techniques used to achieve and maintain the product quality, process and service. Defect repair and measurement of quality indicators.
Tools	<ul style="list-style-type: none"> <li>• Quality Standards</li> <li>• Metric Development</li> <li>• Quality Checklists</li> <li>• Automated Testing</li> <li>• Burn Down Charts</li> <li>• Acceptance Criteria</li> <li>• Sprint Review</li> <li>• Daily-Stand-up</li> </ul>	<ul style="list-style-type: none"> <li>• Assessment of Metrics</li> <li>• Frequent Delivery</li> <li>• Co-Location</li> <li>• Daily-Stand-Up</li> <li>• Sprint Review</li> <li>• Requirements traceability</li> <li>• Testing or Code Inspections</li> <li>• Quality Product Reviews</li> <li>• Quality Process Audits</li> </ul>

## 4.5 Quality Control

Controlling and Monitoring is one of the five project management process groups in the Project Management Body of Knowledge (PMBOK) from the Project Management Institute (2004). The PMBOK definition of Quality Control is the “monitoring the specific project results to determine whether they comply with relevant quality standards and identifying ways to eliminate causes of unsatisfactory performance.”

Traditional project management is very concerned about observing and measuring actual project performance against the plan and fixing it if it varies. Variance from the plan is considered bad unless it is approved by Change Management. Project

monitoring is observation part of this and project control is the corrective action taken as a result. Typically the project manager tracks factors such as cost, quality and effort.

Agile provides more opportunities to monitor what is going on than traditional methods and hence offers more effective opportunities to intervene. The main Agile monitoring technique is to track what software has been incrementally delivered. This is both easier and has much more impact with the customer. The Product Demonstration at the end of a sprint is where the product increment is show cased. With Agile, even though quality is the focus from the very beginning in an agile project, we still seek to validate outcomes and formally track the quality of the product we are building.

### 4.5.1 Monitoring Controls

Agile monitoring and control comes in a variety of forms:

- Frequent delivery
- Colocation
- Daily Stand-Up Meeting
- Sprint Review Meeting
- Requirements Traceability
- Automated Testing
- Releasable Software in every Sprint
- Code Reviews

#### 4.5.1.1 Frequent Delivery

Frequent delivery is key to Agile. Three of the principles behind the [Agile Manifesto](#) relate to frequent delivery of software:

Working software is the primary measure of progress.

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

The best evidence that a software project is on track is working software, preferably deployed to production. There is working software at the end of each sprint – this is what

is reviewed in the Sprint Review Meeting. And there is working software on production during every Release; at least once at the end and usually more often.

Frequent delivery allows the Service Manager to declare a product “done” at any time or the Business Sponsor to declare the project finished at any time.

#### 4.5.1.2 Co-Location

Two of the principles behind the Agile Manifesto are:

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Business people and developers must work together daily throughout the project.

Agile teams try to promote face-to-face communication and to reduce the channels of communication in general. A key way to do this is to be co-located in the same area. Two people sitting next to each other are more likely to discuss an issue than the same two people sitting 20 feet apart in separate offices. The further the distance and the greater number of physical barriers then less likely the discussion will happen at all. Sitting together also makes it easy for several people to cluster around the same screen, if that is needed, and for a group to convene in front of the project board

#### 4.5.1.3 Daily Stand-Up Meeting

The Daily Stand-Up Meeting is focused, short, and daily. The Scrum Master facilitates the meeting. The idea is to check the team’s progress towards the sprint goal and highlight any impediments. The meeting is held daily, at a regular timeslot, and lasts at most 15 minutes; the meeting is a key part of the Agile Lifecycle / Agile Heartbeat. The format of the meeting is predictable, with each team member answering three questions:

- What have you achieved since the last Daily Team Meeting?
- What will you achieve between now and the next Daily Team Meeting?
- What is getting in your way?

The meeting is not about problem solving or discussion. Longer conversations need to be taken off-line. The Project Manager logs the impediments. Some of these can be solved immediately but other need to be fed into Agile Risk Management.

During the daily-stand-up, teams are constantly tracking burndown to see how well the team is doing against the backlog. Within the iteration, the team tracks task progress to make sure that they can deliver on their commitments. Agile teams also track defects, defect status, and test trends. All this gives the team a way to continuously control the project quality.



#### 4.5.1.4 Sprint Review Meeting

Agile teams review features with their customer(s) as they are completed. They do formal product demonstrations and retrospectives at the end of every iteration. These processes allow the team to control, not only the quality of the emerging product, but also of the processes they are using to deliver that product.

The Sprint Review Meeting checks both progress on the product and progress of the process. As a result the Sprint Review Meeting usually has two parts:

- Product Demonstration
- Retrospective

The Agile Team, and specifically the Service Manager and any stakeholders, attend the Sprint Review meeting. Other stakeholders are welcome to the Demonstration but are normally excluded from the Retrospective. For more information on sprint reviews, refer to [Agile Process 202: Sprint Review](#).

The **Product Demonstration** allows the service team to celebrate delivering some software, allows the Service Manager to formally accept the software, and for the service team to discuss what's next. The service team is meant to deliver working software every sprint and the demonstration is to look at the most recent product increment. The product demonstration will spark conversations about further work to be done which will get recorded as User Stories in the Release Plan and/or be reflected in Tasks in subsequent Sprint Plans. This is a key part of Agile as the service manager and stakeholders have early chances to provide feedback to the team and a chance to adjust the product vision based on how the product is evolving.

The **Retrospective** is an opportunity to reflect on the service team's performance and look at ways to improve. For more information on the retrospective, refer to [Agile Procedure 200: Sprint Retrospective](#). Retrospectives cover the following:

- Set the stage: Introductions and Agenda
- Review: Review action points from previous Retrospectives
- Gather data: Timeline – What significant events happened when?
- Generate insights: Things that went well, went badly, things we learned, novel ideas, and things to thank people for (“bouquets”)
- Action plan: Actions to do to improve the process in future – Which ones to actually do – Who will do them – When
- Close: Summarize actions

The service team has to decide which of the action points to implement. Few teams have the capacity to implement all suggestions immediately, so they will only address a subset each time, possibly only the most pressing issue. These process improvement action points should act exactly like User Stories, i.e. they get estimated and go on the Release Plan. In the short term, work on process improvement reduces the amount of work that can be done on new development.

#### 4.5.1.5 Requirements Traceability

Requirements traceability is concerned with documenting the life of a requirement and providing bi-directional traceability between various associated requirements. It enables users to find the origin of each requirement and track every change that was made to this requirement.

In the new world of Agile, requirements are documented through the creation of user stories that live within a specific feature called an epic. These user stories will document the business and functional needs of CWDS and serve as the Product Backlog of the work to be completed for each service team.

Requirements traceability ensures that the business need is tied to epics/features, along with the user stories contained within them, that have been developed based on the previously baselined requirements and approved Business Process Packages (BPPs). Should business needs change, the Agile methodology embraces this and provides traceability through the use of Pivotal Tracker along with the requirements gap analysis being performed throughout user story development/refinement. The user stories will be tied to artifacts such as test cases, providing further traceability. The goal of tracing is to ensure that requirements (and ultimately, solution components) are linked back to a business objective. In other words, traceability ensures that every requirement has a business purpose, and that no requirement is superfluous. During system development, Quality Control review of the RTM is a key work product and should also be reviewed and approved.

Examples of traceability include:

- Approval FSR/SPR to Solicitation Document
- Solicitation Document to system requirements
- System requirements to software requirements
- Software requirements to high level design
- High level design to detailed design
- Detailed design to code
- Software requirement to test case

The Requirements Manager is responsible for conducting traceability on the CWS-NS requirements. The results will be communicated to the Quality Manager and the Project Team to determine the appropriate actions for areas of non-compliance.

Refer to the [CWS-NS Requirements Management Plan](#) for more information.

#### 4.5.1.6 Automated Testing

Agile teams don't wait until the end of the project to test, when they have the least amount of time to actually fix a problem, or respond to a change. Service Teams should know at all times the health of the project, if the team is burning hot, if defects counts are trending up or down, how well they are resolving issues, and if those issues are becoming impediments to getting new product built.

The advantage of automated testing is that we know the health of the product in real time. We are able to measure and track defects and get them resolved as soon as they are introduced into the build. Manual testing, in parallel with the automated testing, gives a more intuitive way to exercise aspects of the code that are difficult to automate.

For example, automated unit tests express the internal behaviors of the software. The total suite of automated unit tests become regression tests and are used to verify that the internals of the software continue to function as designed after subsequent changes, meeting all quality standards. Agile teams aim for 100% pass rate for automated unit tests and system tests. More mature teams write the tests before they write the code in Unit Test Driven Development. Refer to the **CWDS Test Plan** for more information.

**4.5.1.7 Releasable Software in every Sprint**

Each sprint is meant to result in releasable code (or as previously described, in a potentially shippable increment [PSI]). Although meeting the Service manager’s expectations is a priority, it is not the only criteria. Releasable software:

- Meets the Service Manager’s expectations given features asked for and the sprints completed so far
- Meets agreed coding standards
- Meets best design for the currently implemented features
- Is easily maintainable
- Has been tested to the satisfaction of the team and relevant stakeholders

**4.5.1.8 Code Reviews or Inspection**

The “Two Pairs of Eyes” approach provides a peer review of code to validate that it follows agreed coding standards, conforms to design guidelines and is easily understood by developers other than the author. This can be either through pair programming or more traditional code reviews/walkthroughs. Although not mandated by any of the Agile approaches, some service teams may opt to collect metrics about the quality of the code. Examples are code-coverage (amount of code covered by unit tests), conformance to maintainability design principles (e.g. Lack of Cohesion of Services, Normalized Distance from the Main Sequence), and language-specific metrics.

**4.5.2 Data to Use in Monitoring**

Agile projects throw up a lot of data for the Agile Project Manager or scrum master to use in controlling, or steering, the project. A few possible interventions are listed below:

**4.5.2.1 Manage New Risk**

Managing new risks includes identifying, addressing, and eliminating sources of risk before they become a threat to the project. It is important to note Agile’s impact on common project risks:

Common Risk	Agile’s Impact on Risk
Feature Creep	Reduce

Requirements or Develop Gold-Plating	Reduce
Shortchanged Quality	Reduce
Overly optimistic schedules	Reduce
Inadequate design	Possibly Increase
Research-oriented development	Reduce
Friction between developers and customers	Reduce

Good Agile Project Management reduces risk by directly addressing several of these common risks. Unfortunately, Agile potentially increases some of the common risks. On balance Agile reduces more risks than it introduces ... if you do it right.

#### 4.5.2.2 Decrease Scope

At the end of each sprint you should review the release plan to whether the project scope has to change based on the actual Velocity. Often the team finds that the actual Velocity is lower than hoped. In this case scope must decrease by moving User Stories out-of-scope in through the CWDS change management process.

#### 4.5.2.3 Increase Scope

If actual Velocity is higher than anticipated and your service team is churning out more functionality that you thought possible, you can increase the project scope by adding User Stories through the CWDS change management process.

#### 4.5.2.4 Re-Plan the Sprint

If service teams are having trouble meeting their commitments in the sprint, they should re-plan and re-prioritize the remainder of the sprint during backlog refinement exercises. The idea is to get maximum value from the remaining time. There is an overhead in doing a re-plan but this is worth it if your team can't do everything and so has to make trade-offs.

#### 4.5.2.5 Increase Staff or Increase Velocity

If it becomes clear that the service team can't deliver all of the hoped for scope in the time available, it is tempting to add people to increase Velocity. This is contrary to the assumptions of Agile – Agile puts Cost ahead of Scope – and is contrary to an axiom of computer Science that adding people to a failing project doesn't help.

On the other hand, if it is early enough in the project then adding people might help. But don't expect an immediate uplift in Velocity when the new staff arrive. New staff have a learning curve and the old hands have to help them through it, so, in the short term, you may well see Velocity decrease. The important thing is to keep measuring Velocity each sprint and adjust your Release Plan accordingly.

#### 4.5.2.6 Extend the project or add new sprints

If the service team can't deliver the desired scope, another strategy is to extend the project by adding sprints at the end. However, note that this is contrary to an assumption of Agile – putting Time ahead of Scope – but more pragmatically the high priority User Stories are scheduled early and so the last sprints might not add much value. But the customer may be relieved by having the apparent option to get everything they originally wanted.

### 4.6 Improve Quality

PMBOK® defines quality as “*the degree to which a set of inherent characteristics fulfill requirements*”. The discipline of quality management complements project management with a focus on customer satisfaction, prevention of defects over inspection, management responsibility, and continuous improvement.

In Agile, quality improvement refers to anything that enhances an organization's ability to meet quality requirements. It covers product improvement, process improvement and people based improvement. Organizationally, a continual improvement process is an ongoing effort to improve products, services, or processes. These efforts can seek "incremental" improvement over time or "breakthrough" improvement all at once. Processes are constantly evaluated and improved in the light of their efficiency, effectiveness and flexibility.

The concept of Continuous Integration is about maintaining quality all the time, throughout the project. For example, it can involve automatically integrating and running a regression test every time somebody does a code check-in. This is likely to happen several times a day. Running an automated regression test frequently means defects are highlighted soon after they are introduced (i.e. when the build goes Red, i.e. fails). The team's top priority is to get the build Green again.

After the actual level of quality has been established (through Quality Assurance and Control), the deliverables produced and the processes executed should be compared to the quality standards that have been established and quality improvement actions should be implemented as necessary. The level of quality achieved and the preventative or corrective actions undertaken should be communicated to the Project Manager for consideration and the project plan and schedule adjusted accordingly if applicable.

One of the principles behind the Agile Manifesto is:

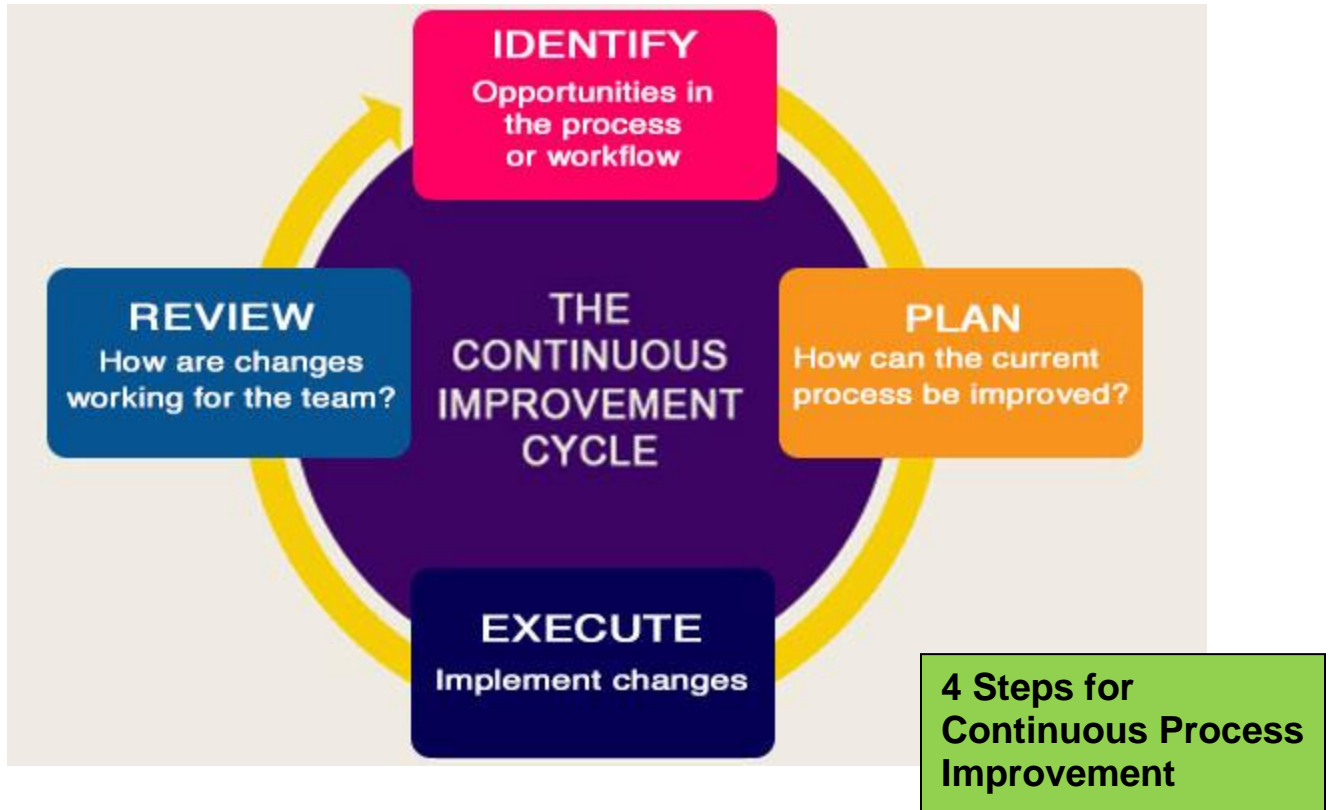
At regular intervals, the team reflects on how to become more effective, then tunes and adjust its behavior accordingly.
--

Retrospectives are the mechanism most service teams use for reflection. During a Retrospective the team looks at how well the sprint went and what they can do different. The high priority changes become User Stories to go into the Release Plan for implementation. Often this is the process to implement new processes.

When looking at how quality can be continuously improved, quality management must include the steps below in Figure 6 for identifying the opportunity, planning the

improvement, executing the improvement and a continuous review of the improved quality standard or process.

Figure 4 – Steps for Continuous Improvement



## Appendix A: Terms and Definitions

Term	Definition
Acceptance Criteria	Pre-established minimum standards or requirements that a project or product must meet before deliverables or processes are accepted.
Backlog Refinement	Backlog refinement is the act of the team coming together to review/refine the stories to bring them to a point of being ready to take into sprint for development. The story and the acceptance criteria are reviewed/refined and validated.
Burn-Down Chart	A Burn-down chart is a chart showing how much is work remaining in a sprint. The outstanding work (sprint backlog) is often on the vertical axis, with time along the horizontal. It is useful for predicting when all of the sprint work will be completed.
Checklists	A list of specific quality criteria to evaluate a quality product review or a quality process audit.
Continuous Process Improvement	An ongoing effort to improve products, services, or processes that seek "incremental" improvement over time or "breakthrough" improvement all at once.
Definition of Done	A global definition of what a team needs to complete for their product or increment to be ready for release. For a development team, this usually encompasses items like: code is checked in; code is merged; unit tests are written, executed and pass.
Epic	An Epic is a very large user story that is eventually broken down into smaller stories; epics are often used as place holders for new ideas that have not been thought out fully. Epics may cross multiple sprints, but should not cross releases.
Product Backlog	The product backlog is the requirements for a system, expressed as a prioritized list of product backlog items. These include both functional and non-functional customer requirements as well as technical team-generated requirements.
Quality Assurance	An executing process that is necessary to continuously improve activities and processes to achieve quality. In Agile, quality assurance means ensuring that software is working right. It includes planning activities to demonstrate quality.
Quality Control	A monitoring and control process that ensures every deliverable and work product is measured and tested to ensure results conform to quality standards. In Agile, quality control focuses on monitoring progress though velocity and tracking what software has been incrementally delivered. It involves implementing the plans made during quality assurance activities.
Quality Criteria	Characteristics of a product or process that determine whether it meets the needs of the customer.
Quality Improvement	Adjusting a process for continuous corrective action or process improvement. In Agile, quality improvement is accomplished through sprint retrospectives.
Quality Management	The application of a quality management system in managing a process to achieve maximum customer satisfaction at the lowest overall cost to the organization while continuing to improve the process.
Quality Metrics	Parameters to quantitatively assess a project's level of quality, along with the processes to carry out such a measurement. Metrics outline the standard that work will be measured against.
Quality Planning	Determining a plan for quality by documenting quality standards and framework. In Agile, quality planning tells us what we expect to do.

Term	Definition
Quality Policy	An organization’s general statement of its beliefs about quality, how quality will come about and its expected result.
Quality Process Audit	A quality control process conducted by the Quality Manager to determine whether project activities comply with the project’s quality policies, processes, and/or procedure, if the process is effective and efficient, and whether the appreciate controls are being applied.
Quality Product Review	A quality assurance process conducted by the Quality Manager and various review members to verify the proper executions of product quality control and follow up actions of non-conformance with documents and deliverables.
Peer Reviews	Informal document or process execution reviews conducted by a group of peers that are knowledgeable and skilled in the subject matter at hand.
Potentially Shippable Increment	Results that are completed to a high degree of confidence and represent work of good quality that is potentially shippable to end customers at the end of a sprint. Being potentially shippable does not mean the results will actually be delivered to customers. Shipping is a business decision; potentially shippable is a state of confidence.
Release Plan	A release plan is an evolving flowchart that describes which features will be delivered in upcoming releases. Each story in a release plan has a rough size estimate associated with it.
Retrospective	The Retrospective meeting is held after the Sprint Review meeting. Its purpose is to provide the Team an opportunity to learn, and therefore improve, from the experience of the just-concluded Sprint.
Service Team	The Team is a self-organized and cross functional group of people who do the hands-on work of developing and testing the product. They are responsible for producing the product, and must also have the authority to make decisions about how to perform the work.
Sprint	The basic development cycle for a project and most often is 2-4 weeks; however, different organizations and projects select Sprint lengths that best meet their needs. Sprints are also called iteration. (Iteration is the repetition of a process)
Sprint Backlog	The Sprint Backlog is the set of Product Backlog Items, or PBIs (Stories and Defects) planned for implementation in a Sprint. The items in the Sprint Backlog must be ranked in the desired order of implementation.
Sprint Planning	The Scrum Master facilitates the Sprint Planning Process, which kicks off the Sprint, and is attended by the team members and the Product Owner. The purpose of this process is to select from the Product Backlog those Product Backlog Items (PBI’s) the Team intends to implement in a Sprint.
Sprint Review	The Sprint Review (also called Sprint Demo) is held at the end of the Sprint. The team demonstrates the Sprint’s completed Product Backlog to the Product Owner and other interested parties. The Sprint Review provides the Product Owner a final chance to make a go/no-go release decision, and gives the Team members a chance to show their work.
User Story	A User story describes a desired feature (functional requirement) in narrative form. It usually contains a name, description, screen and external documents, and information about how the implementation will be tested.
Velocity	Velocity is the rate at which team converts items to “complete or satisfied” in a single Sprint and is usually calculated in Story Points. Velocity answers the question: “How much can the team complete in a Sprint?”