

QAM 12003: Fermilab Software Quality Assurance Program

Revision History

Author	Description of Change	Revision Date
William Boroski	Update links and references.	February 2021
William Boroski	Initial release of the Software Quality Assurance Program document to the Quality Assurance Manual.	January 2016

TABLE OF CONTENTS

1.0 PURPOSE, SCOPE, AND EXCLUSIONS	3
2.0 RESPONSIBILITIES	4
3.0 GRADED APPROACH TO SQA.....	5
3.1 Tailoring	6
4.0 SQA PROGRAM ELEMENTS	7
4.1 Lifecycle Activities.....	7
4.1.1 Requirements	7
4.1.2 Design.....	8
4.1.3 Software Configuration Management	9
4.1.4 Development	9
4.1.5 Verification, Validation, Inspection, and Testing	9
4.1.6 Retirement.....	11
4.1.7 Review/Analysis	11
4.1.8 Assessments.....	11
4.2 Procurement.....	12
4.3 Change/Release Management.....	12
4.4 Problem Reporting & Corrective Action	13
4.5 User Documentation.....	13
4.6 Training	13
APPENDIX A -QUALITY ASSURANCE LEVELS	15
APPENDIX B -QUALITY CONTROL MEASURES.....	16
APPENDIX C - GLOSSARY.....	17
APPENDIX D - ACRONYMS	18
APPENDIX E - REFERENCES.....	18

1.0 PURPOSE, SCOPE, AND EXCLUSIONS

Purpose:

The Fermilab Software Quality Assurance (SQA) Program defines the minimum quality assurance requirements for applications used by Fermilab. Through a graded approach, this program ensures development, management, and delivery of reliable applications that meet or exceed established requirements and expectations through adequate planning, performing, assessing and improving.

Software quality assurance is implemented based on the analysis of potential risks should the application fail or not perform as intended. Evaluating each application against potential consequences allows for the implementation of appropriate quality assurance measures and controls.

Scope:

Software is an all-encompassing term that describes all of the non-hardware items associated with the operation and use of a computing system. Software includes operating systems, programming languages, spreadsheets, word processors, databases, and digital media files.

Applications are a form of software developed or configured to perform a specific task or range of tasks. At Fermilab, applications directly support the execution of Laboratory functions, processes, and procedures. They include custom and packaged programs that provide functions such as performing scientific data analysis, calculating shielding statistics, or tracking safety related information. While many tools such as spreadsheets, databases or programming languages are not applications, some solutions developed using these tools may be considered applications.

Fermilab's SQA Program is applicable to all applications used at the Laboratory. The program is focused on ensuring that applications have the appropriate quality control measures in place to ensure that the applications perform as intended against established requirements.

All applications shall be considered for applicability to the SQA Program. Applications that fall under the SQA Program scope will have at least one of the following characteristics:

- Provides an important Laboratory capability or functionality
- Is part of a business process
- Has readily identifiable mission criticality or impact
- Is subject to external compliance reviews (e.g., audits, regulatory reviews, etc.)

For more guidance on determining whether an application should be included in the scope of the SQA Program, see QAM 12090: Software Quality Assurance Grading & Inventory Procedure.

Exclusions and Special Provisions:

Safety Software:

As described in the Fermilab Quality Assurance Program, Fermilab does not employ safety software under the definition of safety software in DOE O 414.1 Quality Assurance.

Collaboration Applications:

Collaboration applications are defined as any application that has been developed by other organizations in collaboration with Fermilab and utilized on-site. This type of application is split into two classifications: data collection applications and data analysis applications.

- Data collection applications are defined as code written to acquire or collect scientific data. Any application that was developed and is used for data collection shall be graded and inventoried under the SQA program.
- Data analysis applications written to analyze scientific data are covered by the American National Standards Institute/American Society for Quality Z1.13-1999; Quality Systems Guide for Scientific Research (ANSI/ASQ Z1.13-1999) and are not required to be graded and inventoried under the SQA program. See section below for further details.

Applications Developed for Generating Scientific Results:

Special provisions for software and applications developed to support the generation of scientific results, including data analysis collaboration applications are specified within the ANSI/ASQ Z1.13-1999. Quality assurance for these types of software and applications is addressed using approaches described within the standard. The standard acknowledges peer review as a primary mechanism for assuring quality in science and encourages the application of sound engineering/scientific principles to the design of supporting computer software to the extent that the risk associated with the scientific research program warrants. See the ANSI/ASQ Z1.13-1999 standard for specific requirements.

Software Programs and Computer Configurations Designed to Operate and Test Experiments, Accelerator Components, and Associated Equipment:

Special provisions for the development, maintenance, procurement, and use of software programs and computer configurations designed to operate and test experiments, accelerator components and associated equipment are specified within the Fermilab Engineering Manual Section on System Design, subsection titled Software. Examples include Programmable Logic Controller (PLC logic), Field Programmable Gate Arrays and embedded software. See the Fermilab Engineering Manual for specific requirements.

ES&H Applications:

Special provisions are in place for the development, maintenance, procurement, and usage of applications related to environment, safety, and health and are specified in [FESHM 2090](#), *Development, Maintenance, Procurement, and Usage of Software Products Related to Environment, Safety, and Health*.

2.0 RESPONSIBILITIES

Information Management System:

- Management and oversight of the Fermilab SQA Program is facilitated by the Information Management System Owner.

- The Information Management System Owner is responsible for overseeing the development, implementation, and maintenance of the SQA Program in accordance with the requirements in the Fermilab Quality Assurance Program.

Divisions/Sections:

- Laboratory Management of Fermilab organizations that develop, purchase, or maintain applications are responsible for ensuring their organizations, including contractors under their control, are following the SQA program outlined within this document.
- All activities requiring a graded approach shall be properly documented by the responsible organization.
- Divisions and Sections are responsible for following proper processes and procedures required by the SQA Program, and for demonstrating compliance to the SQA Program through self-assessment results.

Application Owners:

- Implement SQA procedures for their applications that follow this SQA Program.
- Maintain inventory of applications under their control, assign appropriate QA levels for each application listed in the inventory, and implement appropriate quality control measures to ensure performance requirements are met.
- Review the inventory of applications under their control on a yearly basis to ensure accuracy, and update the inventory as needed.
- Define and document requirements at an appropriate level.
- Document designs and conduct formal design reviews at a level commensurate with the designated QA level.
- Determine an appropriate development methodology to ensure that requirements are successfully met.
- Remove applications from service when applicable and provide notification of removal to the user community.
- Conduct regular assessments throughout the software lifecycle for applications that have been assigned a High QA level, and determine if assessments are necessary for Moderate and Low applications under their control.
- Ensure all artifacts related to an application are under software change management controls at a level commensurate with the designated QA level.

3.0 GRADED APPROACH TO SQA

The graded approach to software quality assurance is based on a system used for managing applications throughout their lifecycle and takes into consideration all of the following factors:

- The relative importance to safety, safeguards, security, and reputation
- Criticality of application functionality and results produced
- Operational impact of failure
- The magnitude of any hazard involved
- The application's intended function
- Expected life of the application

- Dependencies and impact on other applications or systems

To ensure applications are adequately categorized and characterized, a single grading system is used to determine the appropriate level of quality assurance rigor that should be applied to each application. Quality assurance levels are assigned to specific applications based on the severity of potential consequences that could result if an application fails or does not perform as intended. Reasonable judgment should be used by Application Owners when assigning quality assurance levels.

The process for determining appropriate quality assurance levels and quality control measures for applications consists of 3 steps:

1. Determine if applications fall under the scope of the SQA Program, and inventory applicable applications.
2. Determine the appropriate quality assurance level (grade) for each application listed in the inventory. Quality assurance levels are explained in detail in Appendix A.
3. Apply the appropriate control measures based on the assigned quality assurance level. The Quality Control Measures list defined in Appendix B provides guidelines to follow when applying controls based on the assigned quality assurance level.

Following the graded approach, more stringent requirements apply to the “High” quality assurance level than what would apply to “Moderate” and “Low” quality assurance levels.

3.1 Tailoring

Utilizing a graded approach to implement the Quality Control Measures listed in Appendix B allows Application Owners the ability to tailor the degree to which the mandatory controls are applied. Tailoring must be applied by using the Application Owner’s discretion and judgment, and any tailoring must be justified and documented. Tailoring for example will apply in situations where Fermilab is not the maintainer or owner of the application. In these cases application design, and review controls may not apply.

Examples of situations where tailoring may be appropriate include:

- COTS – Commercial Off the Shelf applications may require tailoring because they are purchased. For example, documenting requirements and design would not be required for the application, but testing of the application would apply.
- Collaboration Software – May require tailoring because the applications are received already developed and ready to use, much like COTS. Again, documenting requirements and design would not be required for the application, but testing of the application would still apply.

For specific grading and inventorying procedures see QAM 12090: Software Quality Assurance Grading and Inventory Procedure.

4.0 SQA PROGRAM ELEMENTS

This SQA Program is intended to provide guidance to Application Owners on the development, implementation, and maintenance of applications under their control, to ensure that applications perform as intended. Tools, techniques, methods, standards, practices or conventions are not specified in this SQA Program; however; whenever possible a common set of practices should be established within application areas to ensure consistent implementation of the SQA program.

4.1 Lifecycle Activities

An important aspect of software quality is the software lifecycle of an application. Existing standards cover lifecycle processes and activities in detail. An applicable standard is Institute of Electrical and Electronics Engineers (IEEE) International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC) 12207:2008, Systems and Software Engineering-Software Life Cycle Processes, which contain common practices and specific recommendations for software lifecycle processes. In addition, IEEE 730, Standard- for Software Quality Assurance Processes contains information about forming software quality assurance plans, and IEEE 1012, Standard for Software Verification and Validation defines software verification and validation processes.

Listed here are the most important activities that need to be considered when choosing appropriate practices and selecting an application development process. Applications that are not classified with a High QA level can choose a reduced or simplified process in each of the areas¹. (See Appendices)

4.1.1 Requirements

Documenting requirements is a known way to properly focus and guide application development. Requirements describe the functions that a system must execute or how well the system must perform a function. Requirements guide system design and the creation of test cases. Test cases validate requirements, therefore providing excellent user acceptance criteria. Requirements should be traceable throughout the software development lifecycle. IEEE 830, Recommended Practice for Software Requirements Specifications, provides standards and a template for documenting high-level requirements.

In collaboration with key stakeholders, Application Owners are responsible for defining and documenting requirements at an appropriate level to ensure that the application meets expectations.

¹ See Appendix A: Quality Assurance Levels for definitions and examples of quality assurance levels – High, Moderate, and Low; and Appendix B: Quality Control Measures for the complete list of mandatory, recommended, and optional quality control measures for each QA level.

Requirements Documentation

Applications shall be documented to ensure that their purpose, use and acceptance criteria are clearly understood, and to support ongoing maintenance to ensure the continued functionality of the applications.

Application Owners are responsible for identifying the documents needed to accomplish these objectives and determining the level of control required.

Requirements documentation should include:

- A description of the functions the application is intended to perform.
- Time-related performance issues (e.g. speed, recovery time, response time).
- Attributes of application operations (e.g. portability, acceptance criteria, access control, and maintainability).
- External interfaces including interactions with people, hardware, and other software and applications including user inputs and outputs shall be included for applications graded as High and is recommended for applications graded as Moderate or Low.

All documentation that requires formalized control shall be done in accordance with Fermilab's [Document Management & Control Policy](#).

4.1.2 Design

The design activity is the process of problem-solving and planning for a software solution. Design documents describe how a feature will look, how it will be developed, and how it will be integrated into the full system. Designs must be tied to requirements. Without this tie, the feature being designed cannot be justified. Designs must also take cyber security into account. Design documents provide an excellent source for early review before further development begins.

Application designs should be documented at a level commensurate with the designated quality assurance level. Commercially purchased applications and applications where Fermilab is not the designer do not require design documentation. However, design documentation is required when these types of applications are modified to create specific applications.

Application Owners are responsible for organizing and conducting formal design reviews at a level commensurate with the designated quality assurance level. The results of design reviews should be documented, and corrective actions should be identified and tracked to completion as appropriate.

Application Design and Implementation Documentation

The following elements should be considered when developing design and implementation documentation:

- Major component descriptions of the software or application design (as related to requirements).
- Technical description.
- Description of the prescribed ranges for inputs and outputs. For applications assigned High quality assurance levels, documentation should include full descriptions of user inputs and outputs.
- Design descriptions that can be translated into code. For applications assigned with High quality assurance levels, details shall be at the code module level. For Moderate and Low quality assurance levels, high-level descriptions of external interfaces and major logic structure are adequate.

4.1.3 Software Configuration Management

Software Configuration Management (SCM) refers to a discipline for evaluating, coordinating, approving, and implementing changes in artifacts that are used to construct and maintain applications. SCM enables the management of artifacts from the initial concept through design, implementation, testing, baselining, building, release, and maintenance.

4.1.4 Development

Development is the process of writing and maintaining the source code, but in a broader sense of the term, it includes all that is involved between the conception of the desired application through to the final build of the application, ideally in a planned and structured process. It consists of developing the design output (e.g., source code) using a programming language, application interface, or other form of development tool suitable for compilation or translation into an executable code. The design, as described in the application design description, is used as the basis for the application development and may need to be modified to reflect changes identified during the implementation phase.

Application development may include research, new development, prototyping, modification, reuse, re-engineering, maintenance, or any other activities that result in an application. It also may include integration of multiple artifacts into a whole system as well as testing of the changes or newly developed programs. The requirements and design documents are inputs for developing testing cases. Sufficient levels of detail in the requirements and design should be present in order to develop test cases. Results of code reviews must be reviewed and approved prior to production release.

4.1.5 Verification, Validation, Inspection, and Testing

4.1.5.1 Verification

Verification is the process of evaluating the work-products of a development phase to determine whether they meet the specified requirements.

Verification activities include review, walkthroughs and inspections of

requirements, design, code and test cases. Depending on the level of risk, it may be appropriate to consider involving an independent reviewer in the verification process.

4.1.5.2 Validation

Validation is the process of evaluating applications during or at the end of the development process to determine whether it satisfies the requirements. Validation is completed via testing the application product. Depending on the level of risk, it may be appropriate to consider involving an independent reviewer in the validation process.

4.1.5.3 Inspection & Testing Documentation

Installation and acceptance activities shall be appropriately planned, executed and documented. Applications with High or Moderate quality assurance levels shall have a documented test plan, including test cases, and test results. Additionally, for applications assigned a High quality assurance level, test evidence shall be recorded for all test cases, and all issues identified during acceptance testing must be documented and the disposition of each issue must be tracked. Depending on the level of risk, it may be appropriate to consider involving an independent reviewer in the inspection and testing process.

Inspection and acceptance testing of applications should include objective evidence of the review of application activities, lifecycle documentation, and test reports to ensure that the software:

- adequately and correctly performs all intended functions;
- meets all established requirements;
- properly handles abnormal conditions and events; and
- does not perform any unintended function that, either by itself or in combination with other functions, can degrade the intended outcomes of the software.

Testing may include unit tests, integration tests, system tests, regressions tests and user acceptance tests. Inspections may range from a set of peers conducting a formal documentation review, to a single peer performing a desktop check of the documentation or code listing. User acceptance test results must be reviewed and approved prior to production release.

Inspection and testing documentation should be developed and organized so that it is traceable to the requirements and design, and should contain the results of inspection and testing activities. This includes results of reviews and tests, and summary status of the software. For applications with a High quality assurance level, each of the following documents should be created and reviewed during each phase of software testing: (1) Test Plan, (2) Test Cases, and (3) Test Results and Evidence.

Inspection and testing documentation shall be stored by a method that is in a central, retrievable, and backed up location (i.e. not on an application owner's hard drive/desktop or personal laptop). For software at the lower quality assurance levels (Moderate/Low), the test plan, test cases and test results may be combined into a single document that is reviewed after completion of acceptance testing.

4.1.6 Retirement

When an application is no longer needed, the Application Owner is responsible for removing it from service, updating its state in the application inventory, and providing appropriate notification to the application user community.

4.1.7 Review/Analysis

Application Owners should conduct regular assessments throughout the software lifecycle for applications that have been assigned a High quality assurance level. Assessments should be conducted on an appropriate level for applications assigned Moderate or Low quality assurance levels.

4.1.8 Assessments

Implementation of effective quality assurance methods will be verified through Fermilab's Assessment process as described in the Fermilab Quality Assurance Program. Nonconformities found during this process shall be recorded and tracked through to closure.

Management Assessment

Management assessments are used by an organization to evaluate its own management processes and their implementation in an effort to identify good and noteworthy practices, uncover issues, identify corrective actions, and ensure that the work being performed is satisfactory and in accordance with requirements. Management assessments should be designed to verify that established processes and procedures, requirements and grading are sufficient and accurate.

Independent Assessment

Independent assessments can be performed by internal or external resources. A peer review qualifies as an independent assessment. Internal resources must be appropriately qualified and have sufficient authority to allow for an unbiased assessment. Independent assessments can also include externally imposed audits and reviews of the SQA program.

4.2 Procurement

Applications and/or application services should be procured in a controlled manner to ensure conformance with specified requirements and expectations of the purchaser. Acquisition documents for commercial software and applications should clearly state or reference requirements for the application being acquired. Acquisition documents and Statement of Work for contracted development and software services should identify all documentation, plans, and procedures to be supplied by the vendor. This vendor supplied documentation for software services/code development must be based on the QA level of the software and follow the mandatory/ recommended documentation as outlined in Appendix B. Documentation covered under confidentiality or nondisclosure agreements is exempt from the list of mandatory/recommended documentation noted in Appendix B.

4.3 Change/Release Management

During the application lifecycle, applications typically experience a number of changes as new features are added and issues are addressed. Change Management is necessary to ensure that changes are evaluated and controlled in a manner commensurate with the QA level. In particular, Application Owners shall develop, specify, or identify Change Management processes to ensure that changes are recorded, evaluated, prioritized, authorized, planned, tested, and implemented in a manner that is commensurate with the QA level of the application.

The Change Management process should establish how change requests are generated, reviewed and approved, as well as acceptance criteria for the end result. The Change Management process should also define the authority for determining changes in distribution modules such as releases, service packs or patches. The required documentation at each stage of the Change Management process, as well as the designated decision-making mechanisms should be directly commensurate with the application quality assurance level.

Where appropriate, a mechanism should be established to facilitate traceability of resulting code or configuration changes to the initial change request. It is also advantageous to provide traceability with the change request approval, implementation timeline, and evaluation of end results. Establishing such a mechanism reinforces the relationship between change management and configuration management. A traceability matrix must be updated during each phase of the project to ensure all requirements are accounted for.

The Release Management process should establish a method for packaging, testing and delivering new or changed components into the production environment with minimal disruption to existing service. A release must be uniquely identified with a version number.

A key component of Release Management is version control. Version control is necessary to ensure that the proper version of an application or developed component is being used. The required level of software version control and management varies by application quality assurance level, and it is the responsibility of the Application Owner to designate and/or document the chosen methodology and approach. Software version control must be established prior to the application being placed into production use.

4.4 Problem Reporting & Corrective Action

Application Owners are responsible for collecting and recording problems, and ensuring corrective actions are identified and tracked through to completion.

Incident reporting for applications tracked by the Fermilab Service Desk shall use the service desk application to report incidents and track corrective actions through to completion following Fermilab IT Service Management processes.

4.5 User Documentation

User documentation should include the following:

- User instructions that include an overview of the application's purpose and functionality, description of the user's interaction, and description of any required training.
- Input and output specifications.
- Input and output formats, where appropriate.
- Description of system limitations.
- Shall be under version control.

The type and detail of user documentation will vary depending on an application's function and quality assurance level assigned. Documentation shall consist of basic operations and an explanation of common errors. For all cases, user documentation may be in the form of online help.

4.6 Training

All personnel associated with the development, acquisition, configuration, management, use, oversight, and retirement of Fermilab applications shall have the appropriate training necessary to perform their assigned job per the Fermilab Quality Assurance Program, either through experience, classroom, or on-the-job training.

Fermilab line management is responsible for establishing job requirements and for ensuring personnel receive the appropriate level of training necessary so that job competency and compliance is maintained.

Any employee performing quality assurance activities, including verification, validation, and auditing should be qualified through experience and/or training.

Application Owners, any employee developing applications, or anyone responsible for determining appropriate quality control measures for computer applications under their control is required to take TRAIN Course Code: FN000510 Introduction to Fermilab's Software Quality Assurance Program.

APPENDIX A –QUALITY ASSURANCE LEVELS

Quality Assurance Level	Potential Consequences if an Application Does Not Perform as Intended
High (If one or more apply)	Causes injury
	Causes an evacuation
	Causes environmental hazard including medium/high potential for reportable radiological release, groundwater contamination, or regulatory violation
	Causes a significant disruption in laboratory operations or business operations
	Causes significant disruption of an experiment or program, or has significant impact on a contractor or Department of Energy (DOE) mission or program
	Compromises data integrity: total loss of or severe reduction in data quality, experimental data or equipment output
	Causes a release of DOE sensitive information
Moderate (If one or more apply)	Causes environmental hazard including low/small potential for reportable radiological release, groundwater contamination, or regulatory violation
	Causes minor program downtime
	Causes a minor disruption in laboratory operations or business operations
	Incurs a minor loss of experimental data or equipment output
	Causes minor disruption to an experiment or program, or has minor impact on a contractor or DOE mission or program
	Causes public release of information not authorized by DOE for public release
	Causes minor reduction in data quality or equipment output
	Can lead to compromises in systems or can cause the release of passwords or credentials that can lead to compromise of such systems
	Can lead to compromises in systems with personal identifiable information (PII).
Low (All should apply)	Does not cause a worker safety-hazard
	Does not result in any evacuation
	Does not cause environmental hazards, causes no environmental impact
	No program downtime
	No/Negligible disruption in laboratory operations or business operations
	No/Negligible loss of experimental data or equipment output
	No adverse public impact
	Causes loss of information that is authorized by DOE for public release.
	No unplanned release of information to the public. Negligible reduction in data quality or equipment output

APPENDIX B –QUALITY CONTROL MEASURES

M = Mandatory

R = Recommended

O= Optional

Control Measure	High	Moderate	Low
Documentation created to design, develop, and maintain the application.	M	M	R
All external interfaces identified, documented and analyzed including user inputs and outputs.	M	R	R
High-level descriptions of major logic structure.	M	R	R
Results of code reviews must be reviewed and approved prior to production release.	M	R	O
Appropriate Test Plan, Test cases and Test Results documents created.	M	M	M
Test evidence is recorded for all test cases.	M	O	O
Each issue identified during inspection and acceptance testing, and its disposition, shall be documented.	M	O	O
Inspections and review documentation created and controlled during each of the lifecycle phases and at the end of the development cycle.	M	R	R
User acceptance test results must be reviewed and approved prior to production release.	M	R	O
Acquisition documents created for procurement of application or software services.	M	M	M
Change Management practices in place to ensure that changes are recorded, evaluated, prioritized, authorized, planned, tested, and implemented per the process.	M	M	R
Application source under version control prior to being placed into production use.	M	M	M
A Traceability matrix must be updated during each phase of the project to ensure all requirements are accounted for.	M	M	R
Collect and record problems, and ensure corrective actions are tracked through to completion.	M	R	O
User documentation under version control.	M	R	R
Documentation of basic operations and explanations of common errors.	M	M	R

APPENDIX C – GLOSSARY

Application – A form of software developed or configured to perform a specific task or range of tasks. An application can be a complex system with many components/programs, each of which may have a different QA level. In this case, the application should be classified as the highest QA level of its components, recognizing that each component may have different controls based on its individual QA level. In the context of this document, software such as firmware is considered an application.

Application Owner - The individual or group with the responsibility to ensure that the program or programs that make up the application, accomplish the specified objective or set of user requirements established for that application, including appropriate security safeguards.

Artifact – Any element created from the software development lifecycle, including but not limited to documentation, test plans, workflow diagrams, data files, and scripts.

Collaboration Applications - Any application that has been developed by an institution in collaboration with Fermilab and utilized on-site. This type of application is split into two classifications: data collection applications and data analysis applications.

DOE Sensitive Information - Sensitive unclassified data, such as personally identifiable information (PII), official use only, and unclassified controlled nuclear information require special handling and protection to prevent misuse of the information for inappropriate purposes. (from <http://energy.gov/ig/downloads/audit-report-ig-0818>)

Loss of Availability - disruption of access to or use of information or an information system.

Loss of Confidentiality - unauthorized disclosure of information.

Loss of Integrity - unauthorized modification or destruction of information.

Software – All of the non-hardware items associated with the operation and use of a computing system.

Software Quality Assurance – Processes and procedures that ensure that an application meets or exceeds established requirements.

APPENDIX D – ACRONYMS

Acronym	Description
ANSI	American National Standards Institute
ASQ	American Society for Quality
FIPS	Federal Information Processing Standards
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IQA	Integrated Quality Assurance
ISO	International Organization for Standardization
ITIL	Information Technology Infrastructure Library
ITNA	Individual Training Needs Assessment
SQA	Software Quality Assurance

APPENDIX E - REFERENCES

- The Fermilab Quality Assurance Program (location: <https://esh-docdb.fnal.gov:440/cgi-bin/RetrieveFile?docid=2469>)
- Document Management & Control Policy (location: http://www.fnal.gov/directorate/Directors_Policy/)
- Fermilab Environment, Safety & Health Manual (FESHM) (location: <http://esh.fnal.gov/xms/ESHQ-Manuals/FESHM>)
- Quality Assurance Manual (QAM) (location: <http://esh.fnal.gov/xms/ESHQ-Manuals/QAM>)
- QAM 12090: Software Quality Assurance Grading & Inventory Procedure (location: <https://esh-docdb.fnal.gov/cgi-bin/RetrieveFile?docid=3194>)
- Fermilab Engineering Manual (location: <https://directorate-docdb.fnal.gov/cgi-bin/RetrieveFile?docid=34&filename=20191119%20Engineering%20Manual.pdf&version=7>)
- FEDERAL INFORMATION PROCESSING STANDARDS - Standards for Security Categorization of Federal Information and Information Systems (location: <http://csrc.nist.gov/publications/fips/fips199/FIPS-PUB-199-final.pdf>)
- CHANGE MANAGEMENT ITIL PROCESS – ITIL documentation can be found in DocDB (location: <https://cd-docdb.fnal.gov/cgi-bin/sso/ShowDocument?docid=3530>)