***1.0* *PURPOSE / SCOPE / RESPONSIBILITIES:***

1.1 **Purpose**

1.1.1 This procedure defines a standard for reviewing software changes to custom written software used in the Alcon Huntington Advanced Optical Device Center (AODC) production and manufacturing facilities.

1.1.2 Common programming practices and standards are necessary to produce robust and maintainable software when using multiple vendors and developers. Adherence to these practices and standards is mandatory for ALL software developed by or for AODC.

1.1.3 The purpose is to assure the code being reviewed:

1.1.3.1 Meets or exceeds the software standards in place for them manufacturing site.

1.1.3.2 Is designed to be scalable, easy to change, and can accommodate future upgrades.

1.1.3.3 Provides solutions to related user requirements.

1.2 **Scope**

1.2.1 **In Scope**

1.2.1.1 This procedure applies to custom software applications written to support manufacturing and production operations at the Alcon Huntington AODC facility.

1.2.1.2 The code reviews will attempt to ensure the reviewed software is following Alcon Huntington AODC software standards as outlined in technical report 2589-TR as well as following best practices in software engineering as defined by current and accepted technology trends.

1.2.1.3 Changes to application code related to original or modified user

requirements.

1.2.2 **Out of Scope**

1.2.2.2 Code or software governed, controlled, and maintained by the Information Technology department (Alcon Business Systems).

1.2.2.3 Commercial-Off-The-Shelf (COTS) software packages.

1.2.2.4 PLC ladder logic, robot control code, and any other embedded automation controller code.

1.3 **Responsibilities**

1.3.1 **Code Review Requestor**

1.3.1.1 Typically this includes, but is not limited to …

* Project Leads
* Validation Engineer
* Contractors
* Anyone contracting out Software Services
* SW Engineer/Developer
* Anyone driving validation efforts where a code review is required.

1.3.1.3. Initiates the code review request by filling in pertinent sections of the code review form # xxx-xxx-xxxx.

1.3.1.4 Specifies the relevant user requirements in need of review.

1.3.1.5 Works with the software developer (either Alcon or contractor) to ensure

that any software being validated will undergo a code review.

1.3.1.6 Schedules code reviews as part of a formal project at periodic intervals as

outlined in this procedure.

1.3.1.7 and submit their work for review and feedback in an effort to deliver software that meets or exceeds all coding expectations.

1.3.2 **Code Reviewer**

1.3.2.1 Typically this includes, but is not limited to …

* Contractors
* SW Engineer/Developer

1.3.2.2 Has a detailed knowledge of this code review process.

1.3.2.3 Ensures code being reviewed conforms to Alcon site SW standards.

1.3.2.4 Verifies code meets user requirements specified by the code review requestor as specified in the code review form.

1.3.2.5 When required, conduct code reviews on project code developed by external contractor.

***2.0* PROCESS DESCRIPTION:**

2.1 The frequency and detail of code reviews will be driven by risk-based factors including:

2.1.1 The complexity of the project and supporting code

2.1.2 The experience of the software developer

2.1.3 The performance and history with a contracted software developer

2.2 The following table outlines the frequency schedule to be followed to determine when code reviews are to be performed.

|  |  |
| --- | --- |
| **Developer** | **Frequency** |
| Local Alcon SW Engineer | * Upon demand * Upon request * Required for Object Freeze |
| SW Contractor | * As outlined in project requirements * Upon demand * Required for Object Freeze |

2.2.1 For project work a schedule will be agreed upon between the project lead and the software developer as to the frequency of code reviews performed.

2.2.1.1 Factors that should be considered when defining this schedule may include:

Complexity of the project

Risk factors defined in risk assessment

Location of the development (off-site vs on-site)

Experience of the software developer

Performance history with vendor

2.2.1.2 An example of a code review schedule might be defined as:

|  |
| --- |
| 10% of project work |
| 50% of project work |
| 75% of project work |
| 100% of project work |

2.3 The code review shall be performed by a subject matter expert (SME) other than the original author of the software being reviewed.

2.4 Before a code review is to be conducted, the SME is to review the user requirements associated with the software project in order to determine if the code can support, in a general sense, the requirements outlined.

2.4.1 The code review requestor will provide a list of the user requirements needing reviews as specified in the code review form #xxx-xxx-xxxxx.

2.4.2 As a code review does not involve testing of the software, only a cursory check of user requirement scope will be conducted.

***Example****:* If the user requirements include a certain user interface or multi- threaded approach or technique, the code reviewer can check for the necessary code infrastructure to ensure that the requirement can be supported.

2.5 All applications developed by AODC personnel and their external vendors or contractors must conform to the software standards currently in place and outlined in 2589-TR. Exceptions to this conformance can only be granted by the Alcon lead software engineer or the Alcon software engineer assigned to the project.

2.5.1 Adherence to these standards is a major focus of the software review

2.5.1.1 Use of a set of standard libraries is strongly preferred. Any deviation from the use of the standard libraries must be approved by the lead software engineer.

2.5.1.1.1 Standard libraries have been thoroughly tested, proven, and provide a majority of the usual functionality such as:

User login/Logout

Error Logging

Database access

User security authentication and verification

2.5.1.2 Modification or replacement in kind of any standard library must not be performed with permission from project or lead software engineer.

2.6 The following is intended as a list of code review checks for review and inclusion as needed, depending on project requirements:

* + - user interface
    - security vulnerabilities
    - data Integrity and system security
    - backend database usage
    - file naming conventions
    - class structure standards
    - variable naming standards
    - method naming standards
    - source code layout (brackets, indentation and spacing)
    - commenting standards (class, method, code comments)
    - error handling standards
    - design scope
    - existence of isolated or dead code
    - functionality fits the current design/architecture
    - programming style
    - required library usage
    - shell code paradigm and multithreaded use
    - program initialization
    - network access

2.7 Using the ICE template DD-SYD-039, the SME code reviewer will review the code and complete this document as the official code review report.

2.8 If the code review performed is due to validation activities, the code review report is to be included as an attachment with the validation paperwork.

2.9 If the code review performed is part of a vendor’s code review schedule as agreed upon in the project requirements, a copy of the code review report will be forwarded to the vendor for their immediate remediation of issues discovered in the review.

2.9.1 The next review of the vendor’s code is to begin with a check of the last code review report conducted as a check for the remediation of issues discovered.

2.9.2 If this is the last code review in a vendor’s review schedule and issues are found, a follow up review is to be scheduled until the code is acceptable for delivery to AODC Alcon Huntington.

**3.0 SAFETY:** NA.

**4.0 REFERENCE DOCUMENTS / DEFINITIONS:**

4.12589-TR - Alcon Code Libraries and Programming Conventions Usage

4.2 **Assembly Version** – all PC application software (custom or COTS) has an assembly version number associated. This number should match the build number portion of the Document Control version number and should be reconciled when checking software out of Document Control. The build number should be identical to the software assembly version value.

4.3 **Build Number** – the build number is the software revision number and is used to reflect the version or state the software in terms of maturity, features and bug remediation. The build or version nomenclature allows a sequence-based identifier that will convey the significance of changes between releases whereby the first sequence is changed for the most significant changes, and changes to sequences after the first represent changes of decreasing significance.

This practice permits users (or potential adopters) to evaluate the evolution and level of bug remediation a given software release has undergone.

At Alcon – Huntington custom software applications will specify the version in the

Format:

Rev 9 (Build W.X.Y.Z)

Where 9 - Document Control Revision number  
 W.X.Y.Z - Software build number

Where W - Major Version Number (0 - ~) manually incremented for

major releases, such as adding many new features to the

solution

X – Minor Version Number (0 - ~) manually incremented for

minor releases, such as introducing small changes to

existing features

Y – Build number (0 - ~) incremented as part of solution builds

performed allowing each build to be tracked and tested.

Z – Revision (0 - ~) Incremented for “hotfixes” or patches to

builds released into the Production environment. This is set

to zero for the initial release of any major/minor version of

the solution.

Build Numbering Guidelines:

* For new application, the version number starts with 1.0.0.0
* If the new version contains only bug fixes, increase the revision number so the version number will be 1.0.0.1.

The build number should be identical to the software assembly version value.

4.4 **Code Review** – a systematic examination of computer source code with the intention of finding mistakes, ensuring user requirements are met in general, check for site coding standards conformance, and improving the overall quality of the software.

4.5 **TFS** - TFS is a [Microsoft](https://en.wikipedia.org/wiki/Microsoft) product that provides [source code management](https://en.wikipedia.org/wiki/Revision_control) (either with Team Foundation Version Control or [Git](https://en.wikipedia.org/wiki/Git_(software))), reporting, [requirements management](https://en.wikipedia.org/wiki/Requirements_management), [project management](https://en.wikipedia.org/wiki/Project_management) (for both [agile software development](https://en.wikipedia.org/wiki/Agile_software_development) and [waterfall teams](https://en.wikipedia.org/wiki/Waterfall_model)), automated builds, lab management, [testing](https://en.wikipedia.org/wiki/Software_testing) and [release management](https://en.wikipedia.org/wiki/Release_management) capabilities. It covers the entire [application lifecycle](https://en.wikipedia.org/wiki/Application_Lifecycle_Management), and enables [DevOps](https://en.wikipedia.org/wiki/DevOps) capabilities. TFS can be used as a back-end to numerous [integrated development environments](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDEs) but is tailored for [Microsoft Visual Studio](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio) and [Eclipse](https://en.wikipedia.org/wiki/Eclipse_(software)) on all platforms.

4.6 **GIT** - Git is a version control system for tracking changes in computer files and coordinating work on those files among multiple people. It is primarily used for source code management in software development, but it can be used to keep track of changes in any set of files.

4.7 **Microsoft Visual Studio** - MS VS is an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) from [Microsoft](https://en.wikipedia.org/wiki/Microsoft). It is used to develop computer programs. Visual Studio includes a [code editor](https://en.wikipedia.org/wiki/Code_editor) supporting [IntelliSense](https://en.wikipedia.org/wiki/IntelliSense) (the [code completion](https://en.wikipedia.org/wiki/Code_completion) component) as well as [code refactoring](https://en.wikipedia.org/wiki/Code_refactoring). [The integrated debugger](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio_Debugger) works both as a source-level debugger and a machine-level debugger.

4.8 **Repository** – a file archive where a large amount of [source code](https://en.wikipedia.org/wiki/Source_code) for software is kept, either publicly or privately.  The repository is used by software projects and other multi-developer projects to handle various versions, manage software patch submissions in an organized fashion.

4.9 **Revision –** indicates a Document Control number that increments with each additional validation and subsequent routing of protocols that reference or are related to the software.

4.10 **Software Standards** – a set of guidelines, formats, rules and styles established by a software development installation with the intent of improving interoperability between different software programs created by different software developers.

4.11 **Source Control** – the management of changes to computer programs commonly referred to as source code.

4.12 **Version –** the version of software at Alcon Huntington consists of two parts; a Revision number and a Build number. The Revision number is used to designate iterations of validation and Document Control processing and the Build number is generated by software engineering.

Ex**:** Rev 9 (Build W.X.Y.Z)

4.13 **Version Reconciliation** – to ensure the software being changed is the latest software validated and installed and running on the manufacturing/production floor.

4.14 **VSTS** - Visual Studio Team Services (VSTS) is a cloud service for collaborating on code development. It provides an integrated set of features that you access through your web browser or IDE client.

4.15 **Working Copy –** a local copy of the repository currently owned and worked on by an individual developer.

***5.0 CURRENT CHANGE:***

|  |  |  |
| --- | --- | --- |
| ***Originator:*** | Tim Buches | |
| ***Why is this change being made?*** | Answers CAPA #375735. No CCR needed; changes are for review of current systems only. | |
| ***What is being changed?*** | A new procedure was created for software engineering code review. | |
|  |  | |
| ***Should document be sent to R&D?*** | Yes | No |
|  | If no, why? Within HWV GOA | |
| ***Is this change applicable to processes/products manufactured in Alcon Ireland?*** | Yes | No |
| ***SE / DE:*** | Jim Pickett, SE | |

***6.0 END OF DOCUMENT.***