PHC Imaging

Technical Operating Procedure

Software Engineering Practices for Personalized HealthCare Imaging Algorithms

1. Purpose

Good software engineering practice adherence is required for compliance with Roche Quality Management System policies. This will ensure that our code development is uniform and well documented to keep us within regulations, allowing us to support development work on regulated systems and devices.

The purpose of this document is to provide the teams responsible for the development of Medical Imaging Algorithms, with high-level principles and concepts making up the software engineering practices in the Personalized HealthCare (PHC) Imaging group.

As different subteams within the PHC Imaging group have different processes, coding standards and tools that they may use, this document is intended to serve as a high-level and an overarching Technical Operating Procedure (TOP) across the PHC imaging subteams. Individual product teams may have more specific coding standards, guidelines, processes tailored to their product areas (not documented in this TOP).

2. Scope and out of scope

This document applies to all rule based and Machine Learning (ML) based algorithms delivered within the PHC Imaging group. Within ML based algorithms/models, the scope of this document is limited to locked models (and not continuously learning models).

1. In Scope

\* Rule-based and ML based algorithm development life cycle

\* High-level guidelines for coding standards

\* High-level guidelines on documentation associated with each algorithm

\* Version control process

\* Peer review process

\* Roles and responsibilities

2. Out of Scope

\* Other procedures and methodologies applied in Roche CSV or PMM

\* Coding standards/guidelines tailored to specific product teams

\* As the algorithms are hosted as gears in Apollo-Flywheel Global Imaging Platform (GIP), additional PHC Informatics (IX) processes-practices will apply to these algorithms, these PHC IX practices are not described in this document

\* Testing Management Procedure

\* Change Management Procedure

\* Release Management Procedure

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

3. Definitions

The following terms and abbreviations are used in this document:

Term: Definition

Algorithm/Model: For the scope of this document, an algorithm/model is the application developed by PHC imaging group to analyze medical images

CNN: Convolutional Neural Networks

CSV: Computer System Validation

Rule-based algorithms: These are rule based deterministic algorithms which produce pre-defined outcomes that are based on a set of certain rules

DS: Data Science

GIP/Flywheel: PHC Informatics’ Flywheel based Global Imaging Platform (This is a part of the PHC Informatics Apollo platform)

ML based algorithms: Machine Learning based probabilistic algorithms are algorithms that define their own set of rules that are based on data outputs

Locked Model: Synonymous with “Locked algorithm”. A model that was developed through data-based AI methods, but does not update itself in real time (although supplemental updates can be made to the software on a regular basis).

PHC: Personalized HealthCare

PHC Imaging Group: For the scope of this document, the following PHC Imaging teams together will constitute the PHC Imaging group - Oncology Imaging team, Ophthalmology Imaging team, Neuroscience Imaging team. Note - Digital Pathology (DP) Imaging team has separate processes in place and are out-of-scope of this TOP

PHC IX: Personalized HealthCare Informatics

Product teams: For the scope of this document, product teams include - Oncology Imaging Products team and Ophthalmology Imaging Products team (NOTE - Neuroscience imaging team currently does not have Neuroscience product teams)

Product Candidate: An algorithm that is planned for imminent deployment as a product (to be used in GxP scenarios e.g. in the conduct of a clinical trial, OR being developed as a Software as a Medical Device)

Product: An algorithm that is being used in GxP scenarios e.g. in the conduct of a clinical trial, OR as a Software as a Medical Device

PMM: Project Management Methodology

SaMD: Software as a Medical Device

TOP: Technical Operating Procedure

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

4. Algorithm development life cycle

4.1 Rule-based algorithm development life cycle

Rule-based algorithms developed within the PHC imaging group should follow the above development life cycle; details on each phase and step are described below (more information on development/experimental, qualification and deployment phase is detailed in the “Life Cycle management” subsection below)

Initialization & Preparation Phase

a. Business case evaluation: In this step, the team will identify the unmet need (objective & requirement) in clinical development and/or practice, and translate it to an algorithm development problem.

Development/Experimental Phase

b (i) Design and b (ii) Coding/Implementation: This step will involve evaluating the algorithm requirements and determining the rules; developing a high-level (aka business logic) as well as a low-level-design (aka programming logic) for the algorithm. The output of this step will be a high-quality working implementation of the algorithm ready for testing.

c. Testing: This step will involve unit and integration testing of the algorithm.

Qualification Phase

d. Algorithm release (as a Flywheel gear): This step will involve converting the algorithm into a Flywheel gear.

e. Troubleshooting/Qualification: This step will involve any troubleshooting activities needed. It will also involve performing Computer System Validation, Technical Validation/Analytical Validation.

Deployment Phase

f. Algorithm deployment (CSV/SaMD) This step will involve deployment of the algorithm as a product (this could entail external deployments as well - e.g at the Independent Review Facility (IRF))

g. Algorithm operations and Maintenance Once the algorithm has been released for use it is subject to Change Management processes in the same manner as any other validated system. A GxP compliance Change Management process will be followed.

4.2 Machine Learning (ML) based algorithm development life cycle

ML based algorithms developed within the PHC imaging group should follow the above development life cycle; details on each phase and step are described below (more information on development/experimental, qualification and deployment phase is detailed in the “Life Cycle Management” subsection below)

Initialization and Preparation Phase:

1. Business case evaluation: In this step, the team will identify the unmet need (objective & requirement) in clinical development and/or practice, and translate it to a modeling problem.

2. Data strategy and data acquisition: In this step, the team will identify the types and the amount of data to support the algorithm development to answer the business questions identified in step a; this step will also involve collecting or acquiring the identified datasets, data curation, databasing and determining if additional annotations or labels are needed.

Development/Experimental Phase:

3. (i) Pipeline design and specification

\* Planning: In this sub-step, the team will design and specify the pipeline to take the data source prepared in step b as input, to address the question listed in step a. The team would also ensure the following -

\* Data quality and systematic biases in the datasets are adequately assessed.

\* Training, validation and holdout/test datasets are prespecified. The split needs to avoid any potential information leakage.

\* Performance measurement is prespecified preferably including how the uncertainty is assessed.

(ii) Data preprocessing: preprocessing pipeline will be developed in this sub-step

(iii) Algorithm training: certain algorithm class for instance Convolutional Neural Networks (CNN) is experimented in this sub-step

(iv) Algorithm evaluation: The most promising algorithm based on validation is tested for the performance on the independent test dataset (also called holdout dataset).

Qualification Phase:

d. Algorithm release (as a Flywheel gear) This step will involve converting the algorithm into a Flywheel gear.

e. Troubleshooting/Qualification This step will involve any troubleshooting activities needed. It will also involve performing Computer System Validation, Technical Validation/Analytical Validation (including characterization of the algorithm)

Deployment Phase

f. Algorithm deployment (CSV/SaMD) This step will involve deployment of the algorithm as a product (this could entail external deployments as well - e.g at the Independent Review Facility (IRF))

g. Algorithm operations and Maintenance Once the algorithm has been released for use it is subject to Change Management processes in the same manner as any other validated system. A GxP compliance Change Management process will be followed.

4.3 Life Cycle Management

- Both rule-based and ML based algorithm development life-cycle typically follow an iterative process prior to the Qualification Phase, and therefore may apply Agile development principles (i.e., algorithm requirements may be managed and incorporated into, over multiple iterations)

- Currently, the development life-cycle is divided into 4 phases - Initialization & preparation, development/experimental phase, qualification phase and deployment phase and the phase gates to go from one phase to the next are shown in the figure below

- Initialization & preparation - As described in the sub-section above, this phase will entail identifying the unmet need (objective & requirement) in clinical development and/or practice, and translate it to a modeling problem. This phase will also include data strategy and data acquisition for ML based algorithms

- Development/Experimental phase - This phase will entail algorithm development before converting it into a Flywheel gear.

- Qualification phase - Qualification phase will begin as the algorithms are converted into Flywheel gears.

- This phase will include Flywheel gear algorithms that are also being utilized in non-GxP scenarios/for internal decision making (i.e research and development purposes) or are product candidates (i.e planned for imminent deployment as a product)

- In this phase, product candidates will subjected to the Computer System Validated (CSV) process to be compliant with 21 Code of Federal Regulations (CFR) Part 11, or other similar regulations (e.g. European Union’s Annex 11) before they are utilized in GxP use cases (i.e. use of the algorithm/gear as a part of the Clinical trial)

- For product candidates that are planned to be developed and deployed as SaMD, appropriate processes will be triggered to ensure compliance with ISO 13485

- Deployment phase - This phase will entail a algorithm that is being used in GxP scenarios e.g. in the conduct of a clinical trial, OR as a Software as a Medical Device.

Within the above rule based/ML based algorithm development frameworks, for each algorithm, product teams are expected to also adhere to the following standards (described in the sections below) around these focus areas - coding, documentation associated with each algorithm, version control, peer review process and roles and responsibilities. The expected standards around each of these focus areas are different between development/experimental phase and qualification/deployment phase.

5. Coding Standards

a. Why Coding Standards

Coding standards are a series of recommendations and mandates for a specific programming language that determines the programming style, procedures, methods, for various aspects of the program written in that language. They are a very critical attribute of algorithm development.

A coding standard ensures that all developers writing the code in a particular language write according to the guidelines specified. This provides consistency in the code, makes the code easy to understand, debug and maintain.

Advantages of Coding Standards:

\* Consistency

\* Quality of the system

\* Increases efficiency

\* Minimize the risk of project failure

\* Reduces complexity

\* Maintenance becomes easy

\* Correction of bugs

\* A comprehensive view

\* Cost saving

b. High-level Coding Standards

Appropriate coding standards for algorithm development ensure that the developer creates

the computer programs to implement the rule based/ML-based algorithm that is adequately managed, modularized, and well commented. Also it ensures to improve on such dimensions as readability, quality, scalability and maintainability of the code.

#1

Modularize your design

\* Follow DRY principle: “Do not Repeat Yourself” and aim to automate repetitive tasks.

#2

Write readable code

\* Use understandable naming conventions for identifiers.

\* Keep your line length 80 characters or less

\* Consistently indent your code

\* Use whitespaces around operators and assignment

#3

Comment your code clearly\*

\* Ensure that you do not comment on your code unnecessarily.

\* Limit Comment Length.

\*Here by comment, we mean the inline/single-line/multi-line comments. Recommendations on the stand-alone document files and docstrings are in the next section “Documentation”

#4

Use virtual environments

\* This will avoid any library clashes, since different projects may need different versions of a certain library.

For algorithms that are promoted to qualification/deployment phase, they should also adhere to the following coding standard

#5

Code Refactoring: in this context the term is formally defined as a process that involves editing and cleaning up previously written software code without changing the function of the code at all. The basic purpose of code refactoring is to make the code more efficient and maintainable.

6. Documentation

Documentation is one of the key focus areas in good software engineering practices for rule based/ML based algorithm development to ensure ease of peer review, reproducibility of results and regulatory compliance (for algorithms that are developed into SaMD/GxP use products).

During the development/experimental phase, each algorithm should adhere to the following documentation standards:

\* Documentation associated with the algorithm should be stored in the README file of the version control platform

\* Documentation should consist of information on the following attributes: ​​how the algorithm was created, the training and the test data characteristics, what alternatives were considered, algorithm evaluation strategies, information on algorithm performance, associated analysis plan etc. along with author’s name and UNIX ID

\* Within the code, Python documentation string “docstring” (or similar functions) to be used to add information around the class, module, function or method

For algorithms that are promoted to qualification/deployment phase - Product teams should also consider the use of documentation automation tools like Sphinx, Pydoc, etc. Use of these automated algorithm documentation tools help ensure consistency, accuracy, improve collaboration, save time and money and could also help ensure regulatory compliance as product candidates are converted to products.

7. Version Control

For development/experimental as well as qualification/deployment phase algorithms, product teams should use only enterprise validated code hosting and code versioning platforms for version control. Currently only GitLab (code.roche.com) and GitHub (github.roche.com) platforms are enterprise validated for code storing-code versioning.

Note: For algorithms that are being promoted to qualification phase (to be hosted as gears within Flywheel), the PHC IX team will host the algorithm’s code in the PHC IX designated code repository.

8. Peer Review

For development/experimental phase algorithms, peer review of the code is required in case of collaborative development with other Data Scientists in the same product team.

\* Pull Request (PR) could be submitted after modular enhancement/feature implementation, debugging and testing

\* PR to be initiated frequently and for short lines of code (e.g <2000 lines) to get immediate feedback and prevent code fatigue by reviewer

For algorithms that are promoted to qualification/deployment phase, a peer review process is required before the algorithm can be converted into a gear within Flywheel. The peer reviewer will be responsible for code review, and for ML-based algorithms, verifying the model performance and reproducibility of results. For demonstrating compliance with this requirement, the reviewer will follow the following process

\* Check-out the code from the version control platform

\* Add a note detailing the review process, summarizing any observations and findings, and confirming the completion of the peer review with the completion date, along with their name and UNIX ID in the README file

\* Check-in the code back into the version control platform

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

9. Roles and Responsibilities

This section provides additional details on roles and responsibilities in the context of rule based/ML based algorithm development life-cycle

Algorithm Owner

\* Responsible for Go/No-go decision to promote the algorithm from experimental phase to qualification phase with input from the peer-reviewer

\* Responsible for market, business case and competitive analysis

\* Responsible for long and short term product roadmap

\* This would typically be the Integrated Solutions Product Lead (ISPL) for that algorithms’ therapeutic area (unless specified otherwise)

Algorithm Developer

\* Responsible for developing the algorithm as per coding standards

\* Responsible for end-to-end algorithm design and communication

\* Responsible for verification (e.g. - unit and integration testing) of each iteration outcome

\* Responsible for documentation associated with each algorithm that they developed

\* Responsible for facilitating peer review (applicability of this peer review process is defined in the preceding “Peer Review” section of this document) for each algorithm that they developed

\* Ensures the Design goals – Performance, Modularity, Reliability, Maintainability, Reusability – are met

\* This would typically be the PHC Imaging Data Scientist developing the algorithm

Peer Reviewer

\* Responsible for code review and overall algorithm performance verification

\* Reproducibility of results

\* Responsible for demonstrating compliance with the peer review requirement (i.e. adding a note in the README file confirming the completion of this process; details on this step are defined in the preceding “Peer Review” section of this document)

\* This would typically be a PHC Imaging Data Scientist (has to be a resource different from the algorithm developer) within the same product team

10. References

1. Project Q

\* Good-ML-Practices-and-Code-Quality-for-Imaging-Team-v0.1.1: Summary and background

2. Stage 0 Framework (Ophtha. PHC Program & PHC DS Imaging, with help of Slalom Consultants)

3. PHC IX Coding Standards document (PL ID 22931586)

4. RGITSC Software Development Process - TOP (PL ID 20403924)

5. AI/ML Guidance document for computer system validation of ML systems