Risk Assessment

When editing a DICOM tag it is imperative to ensure only the intended properties of the file are affected. External Python code libraries (specifically pydicom) are utilized which we do not control so it is essential to have a robust QA check protocol in place.

The main risk is that somehow the image is in some way changed other than the ImageType tag.

Once the code has run, the image in eclipse should be checked against the image in PACS ensuring the image resolution; orientation, geometry etc. are identical between the two systems.

# Things to check between images

* Number of slices
* Pixel size
* Geometric length (Measure a repeatable distance e.g. ant to post on a specific slice)
* Origin location (DICOM and user defined if applicable)
* HU/T1/T2/SUV values (Define an ROI)

|  |  |
| --- | --- |
| Risk | Method of mitigation |
| Python Script alters image | Perform QA checks comparing PACS and eclipse:   * Number of slices * Pixel size * Geometric length (Measure a repeatable distance e.g. ant to post on a specific slice) * Origin location (DICOM and user defined if applicable) * HU/T1/T2/SUV values (Define an ROI) |
| Incorrect files are run through program | Code will only change tags if spaces are present in the Image Type Tag. When the code is run it counts and prints the number of altered files – if this is anything other than 0 you should delete then re-export the incorrectly chosen files from PACS |
| Executable file becomes corrupted or damaged in some way | Contact Physics, the Latest version of the code will be available here: <https://github.com/UCLHp/edit-dicom-image-type> The .py file should be downloaded, recompiled and retested. |