IPEM

Institute of Physics and Engineering in Medicine
01904 610821
office@ipem.ac.uk

Fairmount House
230 Tadcaster Road
York YO24 1ES
ipem.ac.uk

# ADVICE NOTE

# In-house development and sharing of software (including spreadsheets and databases) within nuclear medicine departments

**Author:** Dr Matthew D Walker [a]

[a] Medical Physics and Clinical Engineering, Oxford University Hospitals NHS Foundation Trust, Oxford, OX3 7LE, UK.

**Published as:** Walker, M (2023). *IPEM Advice Note: In-house development and sharing of software (including spreadsheets and databases) within nuclear medicine departments*. Institute of Physics & Engineering in Medicine, York, UK.

## Purpose

This document is intended to provide guidance for organisations and employees engaged in the development of software within departments providing nuclear medicine services within the UK. Many such departments make extensive use of in-house developed software in the delivery of diagnostic and therapeutic services. Although the examples provided as nuclear-medicine specific, the topics covered in this document are applicable more widely across healthcare institutions. The guidance here is intentionally high-level, as the substantive detail is provided within other guidance, regulations, and standards to which the reader is referred.

This document has been endorsed by the BNMS Professional Standards Committee.

**Target audience**: departmental managers, technologists, radiographers, medical physicists, computing support staff

## What is Software? When is Software a Medical Device?

Software can be defined as a set of instructions that control what a computer does. Spreadsheets, databases, scripts, and image processing applications are all examples of software commonly found in nuclear medicine departments. Whether the software is also a medical device depends on what it is intended to do, i.e., whether there is an intended medical purpose. For software qualifying as a medical device there are additional regulations. A few examples are given in Table 1.

In either case, there can be regulatory requirements to be met by software used within a health institution. There are a variety of standards and best practice guides applicable in both cases, with a higher expectation for formal development and testing if the software is a medical device.

**Table 1**. Examples of software commonly used within nuclear medicine departments.

| **Software** | **Description** | **Medical Device?** |
|---|---|---|
| Spreadsheet to calculate GFR | A spreadsheet that takes counts as input (e.g. from a sample counter) and calculates a patient's glomerular filtration rate (GFR) | Yes.<br><br>The intended purpose is to calculate GFR, which can affect chemotherapy dosage or whether a person goes ahead with a kidney transplant. Erroneous calculation of the GFR can cause significant health implications (mis-direction of future treatments) so the classification is medium or high risk. |
| Database for tracking radioactive waste | A database that is used to track the radioactive waste present in different bins, to ensure compliance with the Environmental Permitting Regulations and the efficient disposal of waste | No.<br><br>Although not a medical device, the database could be integral to efficient service provision (waste management; safe disposal of waste). |
| Spreadsheet to calculate the volume of F-18 FDG to draw up from a vial | A spreadsheet taking vial calibration data (volume and activity) and applying decay corrections to estimate the required volume to draw up for the a given patient's injection. This volume guides the operator in dispensing, but they do not rely upon it as the activity is measured. | No<br><br>The activity drawn up is measured in a radionuclide dose calibrator and checked against the intended activity. The volume provided by the spreadsheet is only intended to guide the operator, to make drawing up the correct activity more efficient (reducing their finger dose). Although not a medical device, if the software is poorly designed or gives incorrect output it could lead to increased dose to staff and reduce service efficiency. |
| Spreadsheet for SIRT activity prescription | A spreadsheet to calculate the activity of Y-90 SIRT to prescribe for a given target dose, based on the Tc-99m work-up data (e.g. tumour | Yes.<br><br>The intended purpose is to calculate the activity to inject for this therapy, which is a medical purpose (treatment) |

| | | |
|---|---|---|
| | and normal uptake ratios and volumes) | |
| Image Processing Macro to aid diagnosis or inform treatment (e.g. Aladdin code on a Xeleris™ workstation (GE Healthcare)) | A macro or other embedded program that accesses image processing tools from a library, to perform user-specific image processing. Examples: the display of a gastric emptying curve and calculation of emptying rates; the display of SeHCAT images with calculation of 7-day retention; macro to process renal images. | Yes.<br><br>The intended use is for medical purpose, e.g. diagnosis. Although the programming environment or macro capability might be provided as part of a commercial system (which may also be a medical device), the development and sharing of the macro itself must still meet regulatory requirements. |
| Image Processing Macro for software quality assurance | A macro or other embedded program that accesses image processing tools from a library, to perform user-specific image processing. Examples: calculating the FWHM from an image of a point source; calculating uniformity values; calculating detector sensitivity values for QA purposes | No.<br><br>The intended purpose of the software is not a *medical purpose*. The software is not a medical device. There are however risks associated with incorrect software outputs, including the use of equipment clinically when it is sub-optimal. |

## Software as a Medical Device: more information, regulations, best practice

The MHRA have provided a guide to help developers ascertain if their software is classed as a medical device (here)[1].

The regulation of software that is a medical device is a complex area that is currently evolving. New UK medical device regulations (MDR), expected to apply from July 2025 (at time of writing), will place new requirements on in-house manufactured software [2]. Currently, software as a medical device is regulated if it is placed on the market (e.g. transferred from one Trust to another or developed and sold commercially), or used in Northern Ireland where the EU MDR applies. In the rest of the UK, regulation is from the UK MDR (2002) within which software manufactured and used in-house is exempted. Health care practitioners implementing software do however still have a legal duty of care.

An excellent and detailed overview of the requirements for in-house software development, including references the applicable regulations and standards, is provided in an IPEM guidance document (here)[3].

It is notable that even if software as a medical device is not placed on the market, then NHS Digital Information Standards regarding risk management (DCB 0129 / 0160) are legally mandated in England. These standards are also mandatory if the software supports or influences the care of patients in real- or near-real time, and strongly recommended for other software that supports health services. An organisation's Information Technology department are key stakeholders for many software development projects (e.g. for oversight of software security and dependencies), as are medical device managers.

## Best Practice

Medical device or not, regulated or not, software development should follow best practices for software engineering. The whole software lifecycle should be considered within a framework (covered by a quality management system) that is commensurate with the risks from using the software. Many of the principles described in Annex A of the IPEM guidance document[3] can be applied to software being developed in-house but which does not have an intended medical purpose.

Some of the key steps for best practice include engagement with the end-users before starting development, exploring existing (commercial) solutions, defining user requirements, and ensuring adequate documentation.

There are likely to be departmental limitations in terms of computing expertise and long-term support. There are obvious risks if software that is developed and supported by a single member of staff becomes crucial for delivery of the clinical service. Especially, if it is poorly documented and cannot be adequately supported in the long term when the member of staff leaves, or if there is an enforced upgrade (e.g. of the operating system) which might render the software unusable. Independent review and testing, which is a crucial element of best practice, is also missing from software developed in isolation. Software development should be planned with consideration of the complete software lifecycle together with the available support.

## Collaboration and Sharing of Software

Collaboration and engagement with other organisations can be crucial to develop software that is of a high standard and gains user acceptance, ensuring software utilisation for improved efficiency or increased diagnostic accuracy. Developing formal collaborations will require the setup of contracts specifying how data and intellectual property and liabilities are shared.

It can be tempting for those working within the NHS to share their software with colleagues or indeed friends at other Trusts, but this should not be done without appreciation of the fact that in-house developed software is owned not by the individual that created it but by their employer. Transfer of this intellectual property should only be done with agreement from the Trust management and with appropriate indemnity. There is also the question of ongoing support. After transfer of software (and documentation) from Trust A to Trust B, will Trust A provide support for the software and for how long? Or will Trust B be self-sufficient, and if so, will they have a copy of the source code that they are able to amend? If the software being transferred is a medical device, then sharing in this way constitutes “placing on the market” and the regulatory exemption for in-house manufacture and use will no longer apply. For non-medical devices with minimal commercial value, permission might be given for distribution under the GNU General Public Licence or similar.

While the sharing of software or the co-development of software across organisations is in some ways hampered by the above points surrounding IP and liability, this does not prevent the sharing of best practices, user experiences, or advice given via users’ groups or online forums. The sharing of examples of software-related documentation as a means of demonstrating best practice, or to gain critical appraisal

from the community, is unlikely to raise any of the above concerns. In the absence of a contractual agreement or other approval from Trust management, it is advisable to limit the direct sharing to these "harmless" examples of documentation and snippets of working code. Further advice on the sharing of software is available (here)[4].

Those interested in making connections with other developers are directed to the IPEM Communities of Interest (my.community.ipem.ac.uk), which includes a Clinical and Scientific Computing Community, and a Machine Learning Community.

## References


1. Medicines & Healthcare products Regulatory Agency. Guidance: Medical device stand-alone software including apps (including IVDMDs) v1.10e. April 2023 https://www.gov.uk/government/publications/medical-devices-software-applications-apps

2. Medicines & Healthcare products Regulatory Agency. Standard: Implementation of the future regulations. Updated July 2023. https://www.gov.uk/government/publications/implementation-of-the-future-regulation-of-medical-devices-and-extension-of-standstill-period/implementation-of-the-future-regulations

3. McCarthy et al. IPEM Best-practice guidance for the in-house manufacture of medical devices and nonmedical devices, including software in both cases, for use within the same health institution (2nd Ed.) 25/07/2022. https://www.ipem.ac.uk/resources/other-resources/statements-and-notices/guidance-for-in-house-manufacture-of-medical-devices-and-non-medical-devices-including-software-2nd-ed/

4. Chalkley et al. IPEM Advice Notice: Production and sharing of software in a medical context, including spreadsheets, scripting and functional documents. March 2021. https://www.ipem.ac.uk/resources/other-resources/statements-and-notices/advice-notice-production-and-sharing-of-software-in-a-medical-context-including-spreadsheets-scripting-and-functional-documents/



**Date first published:** 15th August 2023

Reviewed by: Robert Ross, Dr Gregory James, IPEM Nuclear Medicine Special Interest Group, IPEM Clinical and Scientific Computing Special Interest Group.

Endorsed by the British Nuclear Medicine Society (BNMS) Professional Standards Committee.

**Next review date:** 1st July 2025