Preliminary Procedure Guidelines on Quality Control of (Medical) Software in Nuclear Medicine

L Romijn, St. Antonius Hospital, Nieuwegein

General

1. Introduction

Software used in Nuclear Medicine can be a regulated medical device just as a gamma camera is, simply because both are regulated by the European Community Medical Device Directive (Directive 93/42/EEC). Explicitly it is amended in Directive 2007/47/EC to clarify that medical software is a medical device; guidance herein and examples can be easily found as not all software used in healthcare has to be regarded as medical software.

If classified as medical software, both manufacturer and user have to ensure that the software complies to requirements and functional needs of the department prior to use in patient diagnosis or therapy. Validation requirements of software by the manufacturer can be found both in CE and FDA directives. Validation of both general purpose (text editor, spreadsheet, etc) and medical software for the user is less obvious. It implies more than testing: it should define responsibilities, describe a software lifecycle, its intended use and documentation thereof, incorporate procedures to minimize risk, and have focus on maintenance and configuration updates. In short the user must ensure the use of the software as good practice and - not in itself - acknowledge it as a source of increased risk to patient diagnosis, therapy or even patient safety on the department.

2. Scope of software as a medical device in Nuclear Medicine

For use in Nuclear Medicine relevant distinction from the Medical Device Directive can be made between "in vitro diagnostic medical device", "active device for diagnosis" and "active therapeutic device". Software incorporated (embedded) in medical devices is outside the scope of this procedure but can be part of the other chapters regarding the "devices". Stand alone software for general purposes is not a medical device when used in a healthcare setting. When used clinically, however, the user has to ensure the performance of the (home made) software, being his full responsibility. This has long been a topic in nuclear medicine. The following table summarizes software often used in Nuclear Medicine departments, according to European legislation.

Table 1. "Medical" classification of typical software used in Nuclear Medicine

Software device	Medical	type	class	Remark
Patient planning	No			
Patient dosimetry	Yes	In vitro diagnostic		
Image acquisition	Yes	Active diagnosis	IIa	Part of other device not necessarily incorporated
Image processing	Yes	Active diagnosis	IIa	
Image report/display	Yes	Active diagnosis	IIa	
ECG interpretation	Yes	Active therapeutic	IIb	Can also be incorporated in the medical device
Speech recognition	Yes	Active diagnosis	IIa	
Information systems (RIS)	No			
PACS	No			Unless e.g data compression is involved
EPR	No			Except active diagnosis modules within EPR
Interfacing modules (exchange of information between two or more)	Yes			If a medical software module or medical device is involved
Telemedicine systems	No			
General purpose	No			
Home Made	Yes/No			Outside this scope

The CE classification (class in Table 1) prescribes the precautions the manufacturer has to make before marketing the product and is related to patient risk involved with the medical software use.

3. Requirements and functional specifications

As "apparatus", medical devices (hardware) have specifications from the manufacturer that can be checked at acceptance. This is however not as obvious the case for medical software. Apart from hardware and software specifications, the intended use and functionality of the software have to be considered. Therefore possibly all user scenarios of the software's intended use, should be formulated as user requirements. They are of major importance for the implementation of the software in the hospital environment, at acceptance and during life time of the software use. Functional specifications of the medical software application should also be formulated and documented. The goal of user requirements and functional specifications differ. Basic goals of the user requirements are to impose constraints on design and implementation :

- the software configuration conforms to its intended use
- the clinical use is effective and efficient
- risks are mitigated
- the risks involved are acceptable compared to benefits
- law and (national) regulations enforced are met

whereas functional specifications of a software application also involves addressing method references and clear predictions of outcome or handling. Formulation of user scenarios is helpful to describe basic goals of user interaction with the software :

- patient outcome (diagnostic or therapeutic performance) is the same or better than before
- department workflow is adequate for software use
- the use of the software is familiar
- the hospital environment is interoperable (compatible) with the software

User requirements of software can easily be the same for two departments, whereas the functional specifications might differ. Typically the IHE (Integrating the Healthcare Enterprise) is a platform that helps specifying functionality from a user point of view. IHE profiles are common in Radiology practice. If IHE functionality is in full compliance with user scenarios (use cases) these profiles can be stated in user requirements and acceptance can be derived from IHE Connectathon results. Otherwise functionality should be tested by the user based on specific scenarios and functional tests (see appendix II for examples).

4. Risks involved in (medical) software use

As already noted, the risk related to software use within healthcare is in itself not a criterion for its qualification as a medical device or not. Software which is intended to create or modify medical information is, however,qualified as a medical device.

The risk of software malfunction in both general purpose and medical software use is defined in various software validation tools. Risk assessment tools used for processes or medical devices can also be applied to software, such as the Healthcare Failure Mode and Effect Analysis (HFMEA), or the well known Good Automated Manufacturing Practice (GAMP as used in Pharmacy). In the following table the category indicates the risk involved with the software use. Not surprisingly, validation of high category (medical) software takes more effort.

Table 2: Risk categories, from low to high, of (medical) software products in Nuclear Medicine by GAMP

Category	GAMP	Description	Examples
1	IT infrastructure	Operating systems Databases Office applications Anti-Virus tools	PACS / RIS
2	No longer used		
3	Non-configured software	Off-the shelf products Software with default configuration	Patient dosimetry Image acquisition Image display ECG interpretation Speech recognition EPR
4	Configured software	Configuration of specific processes Configuration with vendor-supplied scripting	Image processing Interfacing modules (dicom mpps, HL7)
5	Custom software	Developed to meet company regulations Internal application macro's High inherent risk of software use (e.g. ROI's)	Some image processing modules

In the ECRI Institute top 10 of risks involved in health technology 2014, both data integrity failures in EHR and other Health IT systems as well as neglecting change management for networked devices are present, which clarifies the risks involved with the presence or absence of interfacing modules.
Apart from these risks in Nuclear Medicine, all quantitative software packages have to be considered as category 4 or 5 according to GAMP and should be addressed accordingly with user requirements and functional specifications. Software use in default settings represents less risk, like non-quantitative software use.

5. The adjustment of (medical) software

Adjustment of medical software has to be considered in the following cases

- acceptance
- undefined user / administration roles to change configuration settings
- addition or change of procedures
- updates / upgrades (bug fixes)

- updates / upgrades (bug fixes) of interface modules or other connected software
- addition or change of connected hardware
- multi-vendor environment

6. Selection of tests and frequency

Although software Quality control is not new to nuclear medicine, its scope is underestimated. In order to improve this situation the following procedure is suggested.

- User requirements and functional specifications are made for every application in use in the department
- In a risk strategy approach, user risks of the applications are defined, as well as the use of various interface modules
- A Quality Control procedure is introduced in the department to perform user (test) scenarios and measure and minimize risks involved in these procedures

An approach from high to low risk user scenarios of the applications will perform best. To test the user scenarios, specific tests are considered and their frequency should be selected. These specific tests should either confirm user requirements or assess risks involved in software usage by functional testing. Examples can be found in the appendix I and II.

7. Software and system integration

Basically no other procedures for software control are needed for interface modules. Consideration should be on the "experts" that need to be consulted. An example of integration with a clinical document system can be found in http://www.himss.org/ASP/topics_FocusDynamic.asp?faid=295.

8. Archiving and log book

All results of software Quality Control should be documented. At acceptance, the user requirements, the results of the risk assessment, and functional tests performed should be documented and archived also.

9. Miscellaneous

By defining and documenting user requirements, performing a risk assessment and functional tests, quality control of the software is suggested. As quality control of software is relatively new and therefore in many aspects also new to the field of nuclear medicine, enclosed protocols form a means to objectively improve total quality of service. Not new to the field of nuclear medicine however, is the fact that experience over the last 20 years or so, induced a workflow were numerical results of patient studies were related to visual patient outcome, all this in a controlled process of combining professional experience in a multi expertise setting. By the way the patient workflow is structured, a department may and can choose when and how to implement procedures suggested.

10. Abbreviations

CE : Conformite Europeene, CE mark, only medical devices with this mark may be used in Europe (exception can be made only under strict legislation), see : http://ec.europa.eu

Dicom :	Digital Image Communications in Medicine, see : www.dicom.nema.org
ECRI Institute :	A Federal Patient Safety Institute that delivers research information, advice an d safety alerts
EPR :	Electronic Patient Record
EHR :	Electronic Health Record
FDA :	Food Drug Agency, see : www.fda.gov
HFMEA:	Healthcare Failure Mode Effect Analysis, see : www.patientsafety.va.gov/professionals/onthejob/hfmea.asp
GAMP :	Good Automated Manufacturing Practice, see : www.ispe.org/gamp-good-practice-guides
HIMMS:	Healthcare Information and Management Systems Society, see : www.himms.org
HIS :	Hospital Information System
HL7 :	Health Level 7, a non-profit organization developing a framework and related standards for the exchange of electronic health information, see: www.hl7.org
HL7 ADT :	HL7, Admission, Discharge & Transfer messages, typically used when two patient id numbers refer to the same patient (merge) and when the patient name changes (e.g. marriage)
IAEA :	International Atomic Energy Agency
IHE :	Integrating the Healthcare Enterprise, an initiative of both professionals and industry to improve the way computer systems in healthcare share information, see : www.ihe.net
Interface Module :	Software module in between two or more software applications / devices, to exchange data of specific type and format, e.g. HL7 messages from the HIS to the PACS, Dicom messages from a modality to the RIS or PACS
IPEM :	Institute of Physics and Engineering in Medicine, a SIG about NM software quality group reports on software quality
MDD :	Medical Device Directive, European legislation that every European member state has to adapt in its own legislation
PACS :	Picture Archiving and Communication System
RIS :	Radiology Information System
Speech Recognition:	Software application that transforms spoken words into computer text

11. Literature

- Guidelines on the qualification and classification of stand alone software used in healthcare within the regulatory framework of medical devices, European Commission DG Health and Consumer, MEDDEV 2.1/6 January 2012.
- General Principles of Software Validation; Final Guidance for Industry and FDA Staff October 24, 2007, see : www.fda.gov.
- Validation Procedures of Software Applied in Nuclear Instruments, Proceedings of a technical meeting in Vienna, 20-23 November 2006, IAEA-TECDOC-1565 .
- Quality assurance of medical software, Journal Med. Eng. Technology, Cosgriff PS, 1994 Jan-Feb; 18(1); 1-10, see : www.ipem.ac.uk.
- IHE Radiology Technical Framework Supplement 2007-2008, Nuclear Medicine Image Profile NMI with

Cardiac Option.

- HFMEA, see : www.patientsafety.va.gov/professionals/onthejob/hfmea.asp
- GAMP, see : www.ispe.org/gamp-good-practice-guides.
- Top 10 Health Technology Hazards for 2014 (adapted from Health Devices vol. 42 issue 11),November 2013, see : www.ECRI.org.
- Integrating Medical Devices with clinical document systems: a quick start quide, developed by the HIMSS, see : http://www.himss.org/ASP/topics_FocusDynamic.asp?faid=295.

User Requirement Specification

1. Introduction and rationale

User requirements must be formulated SMART (Specific, Measurable, Acceptable, Realistic, Time). They should describe to what performance, engineering and quality standards the software should match. Generally, software requirements will be part of the hospital ICT governance structure; department standards should be added to these requirements.

2. Frequency

At acceptance and at updates or upgrades of the software product or interfaced software applications, depending on the update or upgrade specifications.

3. Method

Software requirements specification is simply a list of all that is needed for project development, and should therefore result from a project team from both manufacturer and customer. An example of requirements that should be documented is given in the following table.

Table 3: Example specifications to include in a user requirement specification. Classification E: essential, W: wanted, or O: optional

Section	Details	Classification
Introduction	Contract status of the document Relation to other documents	M E
Overview	Background Key objectives & benefits (e.g. workload) Main functions and interfaces Applicable requirements (CE, GMP, etc) Other regulations and guidelines (NVNG) Traceability	D D E E E O

Operational Requirements / System Functions	Functions required (e.g. medical exam)	E
	Inputs & Outputs	E
	Calculations & Algorithms	O
	Consistent use of Normal Databases	O
	Plausibility check on entered data	W
	Authorisation mode	W
	Modes of operation (e.g. user, admin)	W
	Quantitative performance requirements	O
	Back up & restore	E
	Access security	E
	User Manual	E
System Implementation Life cycle	Maintenance	E
	Configuration (e.g. default settings)	W
	Logging	W
	Enhancement (e.g. bug, error reporting)	O
	Safety precautions	E
Data Handling Requirements	Definition of (dicom) data (in- & output)	E
	Definition of Patient administration data	E
	Performed procedure in acquisition data	W
	Capacity requirements (e.g. disk space)	W
	Access speed requirements	O
	Recall procedure	O
	Data security and integrity	E
	Conversion/migration strategy	E
	Patient delete of data on request / legal	E
System Interfaces	Roles and functions	W
	Interface to other systems (e.g. PACS)	E
	Interface to other (medical) devices	E
	Interface to Patient Administration system	E
Environment	Physical conditions	W
	Number of licences	O
Constraints	Timescales / Milestones	O
	Compatibility (with current IT infrastructure)	W
	Availability (24/7 or 9-17)	O
	Quality (control)	E
	Procedural constraints	W
	Security	E
	Cost	W
Glossary	Definition of terms	W

4. Requirements and responsibilities

As software installation and configuration requires a project team of participants of various disciplines (*e.g.* ICT, Medical Engineering), and will inevitably be a topic for the department of Nuclear Medicine too, it is obvious to define ones roles in the software lifecycle model. An example is given in the following table, it can be used for both CE-marked software as well as other software applications used on the department.

Table 4: Example of software life cycle involvement regarding user requirements to be considered

SOFTWARE LIFE CYCLE MODEL	Responsible	Accountable	Consulted	Informed
Design	ICT	Manufacturer	Physicist	NM
Implementation	Manufacturer	Med.Engineering	Physicist	NM
Setup / Configuration	Manufacturer	ICT	Med.Engineering	NM
Documentation	Manufacturer	Med.Engineering	Physicist	ICT
Acceptance Testing	NM	Physicist	Manufacturer	ICT
Data Conversion / Migration	ICT	Manufacturer	Physicist	NM
Performance	ICT	Manufacturer		
Maintenance	ICT	Manufacturer	Physicist	NM
Virus Protection	ICT	Manufacturer	Med.Engineering	NM
Logging	Med.Engineering	Manufacturer	ICT	NM
Updates / Upgrades	ICT	Manufacturer	Physicist	NM
Backup / Recover	ICT	Manufacturer	Physicist	NM
Administration	Manufacturer	Med.Engineering	ICT	NM
De-Installation	ICT	Med.Engineering	Physicist	NM

5. Procedure
At acceptance, a document from the project team should reveal all requirements met and specifications that still need to be addressed.

6. Analysis and interpretation
All requirements not met at acceptance should be incorporated in the risk assessment before the software is used. Further agreements should be made with the vendor.

7. Action thresholds and actions
All requirements that lead to unacceptable risk prevent the use of the software.

8. Pitfalls and marginal notes
It should be clear between vendor and user beforehand what to do if software validation fails at acceptance. Temporary use of the former software or other fall back scenarios have to be discussed as part of the contract.

Risk Assessment

1. Introduction and rationale
Tools for accessing risk can be derived from any risk assessment tool, well known are the GAMP, the Fault Tree Analysis (FTA), the Failure Mode Effect Analysis (FMEA) or specifically for Healthcare (HFMEA), or the Failure Mode Effects and Criticality Analysis (FMECA).

2. Frequency
Risks should be assessed at acceptance, when updating or upgrading the software (or its interfaced products) and when software "bugs" are noticed. Review of risks is appreciated as a governance tool, and should be part of the quality control of software.

3. Method
Typically, risk assessment is done by defining an "expert" team, setting the context, defining the process into sub-processes, and per sub-process defining hazards and risks, and quantifying each risk separately, then considering how to manage the risk and define a potential risk that is acceptable. The following table shows some examples of risks to consider.

Table 5: Incidence scoring form of Software related Failures on the department (Example)

nr	Software	Failure	User Cause	Software Cause	Configuration	#
1	Patient planning	Patient ID	Wrongly typed		No Dicom mwl	
2		PatientName		No ADT update	HL7-adt08	
3		Birthdate		No ADT update	HL7-adt08	
		Patient not present	Mutation in planning	Mutation messages not sent	Interface module	
4	Patient dosimetry	Patient ID	Wrongly typed	Mutation messages not sent		
5		Wrong Isotope	Mix up two patients same patient name			
6	Image acquistion	Collimator	Change not noticed in acquisition configuration		No preference	
7		Data loss	Unauthorised configuration change		Acquisition parameters can be changed by user	
8		Data loss		Corruption in Dicom transfer	No Dicom storage commit	
9		Wrong historic images		CD Import	Wrong image numbers in CT series	
10		No historic images	No CD import possible	Private SOP not allowed in PACS	Not configured	
11		PET down	Not noticed "alarm"	Harddisk full	No watermark	
12	Image processing	SUV false	PET Time not checked	Time set	Winter/summer No time server	
13		ROI false	User "stomach" interpretation		Training	
14		MPI AC image wrong SSS		Bug report		
15	Speech recognition	No recognition	Got "cold"	Failure		
16	Patient report	Wrong images	Wrong "fix" or "merge"			

4. Requirements

An incidence scoring of software issues should be part of the department's procedure of software quality control.

5. Procedure

All possible risks, prioritized and, categorized by their analysed effect, should be considered with the people responsible.

Business process

- downtime (unplanned) of a application or system
- too long acquisition times

Product quality

- Noticeable :
  - non-routine acquisition conditions
    - incorrect system time setting
    - incorrect acquisition settings (*e.g.* isotope)
    - incorrect camera preparation (*e.g.* collimator)
    - incorrect exam performance (*e.g.* patient movement)
    - loss of patient data
  - non-routine processing settings
    - incorrect drawing of ROI
    - incorrect attenuation mask setting
- Unnoticed:
  - incorrect outcome of numeric calculations *e.g.* SUV
  - incorrect monitor lut settings
- Unattended
  - false acquisition configuration settings
    - incorrect timing in acquisition software
    - false configuration of SPECT reconstruction parameters
  - false processing configuration settings
    - false SUV calculation configuration

Patient safety

- false administrated dose or isotope to patient
- false image storage in PACS
  - false patient name
  - false patient orientation (*e.g.* L and R interchanged)
- failures in speech recognition software
- failures in PACS viewing synchronisation (*e.g.* images of 2 patients on screen)

6. Analysis and interpretation

Risk analysis and interpretation can be considered for the nuclear 'business' process, the nuclear product 'quality' and patient safety. The risk assessment should formulate the results as:

Accept :	do nothing
Retain :	allocate resources just incase
Transfer :	insurance

Mitigate : do something after the event
Control : do something to reduce risk
Prevent : do something to prevent risk
Avoid : stop doing the action that causes the risk

7. Action thresholds and actions

Thresholds for action must be defined according to various risk classes, the following classes are suggested regarding patient safety:

Class 1a : possible data errors or loss of numerous patient data
1b : possible result errors of numerous patient data
Class 2a : possible data error or loss of data of one patient
2b : possible result error of patient data
Class 3a : possible loss of configuration
3b : possible loss of former results
Class 4a : possible consultation of wrong patient data or results
4b : unauthorised consultation of patient data

8. Pitfalls and marginal notes

Examples of precaution measurements:.

- critical data will always be checked by a second person when entered in a system
- the software checks the entered data on plausibility
- critical data has to be authorised before further use

Appendix I: user Requirement control

Table I, Examples of requirement test procedures

Requirement Test	Purpose	Manufacturer	User	Outcome
Operational Requirements / System Functions	Hardware specification check	Specification	Installed base	Match
	Software specification check	Specification	Installed base	Match
	Restrictions specification check	Specification	*e.g.* no other applications running	Match
	Vendor maintenance			
	Test user defined roles		User / admin	Risk of unwanted configuration change
	Online manual			Online "HELP" functionality
	Performance	Specification		Risk of "lack" of performance
	Backup & Restore	Perform back-up & restore (*e.g.* checksum control)		Full functionality after Restore
System Implementation Life cycle	Maintenance			Prevent "malfunction"
	Safety precautions			Patient Safety
Data Handling Requirements	Dicom conformance			
	Conversion / migration			
System Interfaces	Roles / functions		Unauthorised restrictions	No unauthorised access
	Interface to Patient Administration system			
Environment	Setup and configuration		Check licenses	Match
Constraints	Quality			
	Security			

Appendix II: user Functional tests
Table II, Examples of functional test procedures

Functional Test	Purpose	Manufacturer	User	Outcome
Validation	Clinical validation	Alpha & Beta	Beta testing	
		FDA & CE	Monkey testing	
			Error handling	
	Operability		Use of GUI	Risk on user mistakes
			In connection with other systems	Risk on interface mistakes
			Risk on errors	Risk on mistakes
Verification	Clinical verification		International datasets (IPEM)	International compliance
			Phantom measurements	Historic compliance
			Simulated datasets	Historic compliance
			Prior patient data sets	Historic compliance
			Recall procedure	
Education	Training (e.g. teaching files)		Reproducibility (between users)	Risk in inconsistent usage
Interoperability	Test of software in hospital environment	Installation & configuration	Time synchronisation	Risk on time failures
			Data consistency (e.g. dicom workflow)	Risk on dicom interoperability errors
			Database consistency (ADT messages)	Risk on "loss" of historic patient data
			Interoperability with all department systems	Risk on mistakes with data exchange
			Interoperability external (e.g. CD's)	Risk on mistakes with data exchange
Black Box	Test cases (e.g. historic mistakes)			Prevent known risks

Bug reports	Prevent mistakes	Publication		Preventive actions (*e.g.* procedural changes)
Effect measurements	What will happen if the inevitable ..		Simulate errors	Preventive actions (*e.g.* procedural changes)