DEVELOPMENT OF A MODULAR SOFTWARE PLATFORM FOR DIGITAL X-RAY SYSTEMS

by

Altay Brusan

B.S. in Computer Science, Shahid Beheshti University

M.S. in Mechatronics, K.N.Toosi University of Technology

Submitted to the Institute of Biomedical Engineering

in partial fulfillment of the requirements

for the degree of

Doctor

of

Philosophy

Boğaziçi University

2020

ACKNOWLEDGMENTS

I am sincerely graceful of my mother for her kind supports all the time, my father how encouraged to persuade me to follow up my dreams, and my brother for his advices.

I am thankful of my supervisor, Prof. Dr. Cengizhan Ozturk for his kind supports and great motivation during this work. I would like to announce my gratitude to my team member, friend and boss, Aytac Durmaz for all long journey that we had together. I also thanks Asuman KOLBAŞI for here helps on this work.

I would like to thank TUBITAK (The Scientific and Technological Research Council of Turkey, grant numbers 116E738 and 116E739) and Bogazici University Research Fund (BAP 6003) for the grants that provided us the necessary financial support to complete this work.

ACADEMIC ETHICS AND INTEGRITY STATEMENT

I, Altay BRUSAN, hereby certify that I am aware of the Academic Ethics and Integrity Policy issued by the Council of Higher Education (YÖK) and I fully acknowledge all the consequences due to its violation by plagiarism or any other way.

Name :

Signature:

Date:

ABSTRACT

DEVELOPMENT OF A MODULAR SOFTWARE PLATFORM FOR DIGITAL X-RAY SYSTEMS

Health centers require full-body X-ray imaging in their busy trauma departments. Full-body X-ray scans are required for bone examinations such as diagnosis of osteoarthritis and degenerative changes in the lumbar spine. The most preferred solution is to take digital X-ray images and then to stitch them together via a dedicated software. This approach has limitations concerning imaging time, costs, applied dose, and image quality. An alternate solution that optimizes all these factors for full-body imaging is possible. This device employs Time Delay Integration (TDI) X-ray detector, a new digital X-ray sensor design, with higher resolution and sensitivity in comparison to conventional flat-panel detectors. This thesis introduces a software platform to build customizable X-ray scanners and its first implementation in a device with TDI detector. The software solution for TDI based full-body scanning involves novel ideas and approaches which improve the device performance in terms of the applied dose, quantitative image quality, and applicability in comparison to current available solutions. Although the software solution is implemented first for specific TDI scanner, the mentality behind the platform is to provide a solution for all kinds of X-ray machines. The software platform is a unified extensible architecture for any kind of X-ray machine.

Keywords: X-ray Imaging, Radiography, Medical Imaging Software, Open Source, Full body scanner.

ÖZET

DİJİTAL X-IŞINI GÖRÜNTÜLEME SİSTEMLERİ İÇİN MODÜLER BİR YAZILIM PLATFORMUNUN GELİŞTİRİLMESİ

Tüm vücut X-ışını görüntüleme sistemleri, hastanelerin acil servis ve travmatoloji bölümlerinde sıkça kullanılmaktadır. Bu sistemler ayrıca, osteoartirit ve lumbar bölgedeki dejeneratif değişiklikler gibi problemlerin tanısında ve diğer kemik yapısı tetkiklerinde de yaygın bir ihtiyaçtır. Bunlar gibi geniş bölge ve tüm vücut görüntülemelerinde, çoğunlukla birkaç ayrı X-ışını görüntüsü alınıp yazılım yardımı ile birleştirilerek kullanılmaktadır. Bu yaklaşımın zaman, hasta dozu, görüntü kalitesi ve maliyet gibi sınırlamaları vardır. Son dönemlerde tüm bu faktörleri optimize ederek daha iyi bir performans sağlayan bir çözüm geliştirilmiş olup bu cihaz, geleneksel flat panel detektörlerle karşılaştırıldığında daha hassas ve çözünürlüğü yüksek görüntüler veren, zaman geciktirmeli biriktirme (TDI – Time Delay Integration) metodunu kullanan yeni bir dijital görüntüleme sistemidir. Bu tez çalışmasında, kullanım ihtiyacına özgü tasarlanacak dijital X-ışını sistemleri için bir yazılım platformu geliştirilmiştir ve bunun ilk uygulaması TDI dedektör ile çalışan bir x-ray sistemi için yapılmıştır. İlk olarak TDI tabanlı tüm vücut tarama sistemi için geliştirilen bu çözüm; uygulanan doz, nicel görüntü kalitesi ve mevcut tüm çözümlere kıyasla uygulanabilirlik açısından cihaz performansını artıran yeni fikirler ve yaklaşımlar içermektedir. Geliştirilen bu yazılım, TDI tabanlı görüntüleme sistemleri için tasarlanmış ve denenmiş olsa da yapılan çalışmada benimsenen metodoloji, sadece bu sistemlerle sınırlı kalmayarak, her türlü x-ışını görüntüleme sistemlerine uygulanabilir bir çözüm sunmaktadır. Teknik açıdan, geliştirilen bu yazılım platformu hali hazırda TDI tabanlı sistemler için optimize edilmiş, ancak tüm x-ışını görüntüleme sistemlerine uygulanabilecek birleşik ve genişletilebilir bir mimariye sahiptir.

Anahtar Sözcükler: : X-ışını Görüntüleme, Radyografi, Medikal Görüntüleme Yazılımı, Açık-Kaynak, Tüm-vücut Tarayıcı.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

I_0	Photon intensity at source
I	Photon intensity at distance r
K_{air}	Air kerma
D_i	Data value at step i of time delay integration detector
S_i	Signal value at step i of time delay integration detector
n_i	Noise value at step i of time delay integration detector
μ	Mean value of photon intensity
μ	Mean value of photon intensity
ρ	Material density
σ	Variance of photon intensity
E	Photon energy
h	Planck's constant
f	Photon frequency
cGy	Centigray

LIST OF ABBREVIATIONS

PACS	Picture Archiving and Communication System
DICOM	Digital Imaging and Communications in Medicine
RIS	Radiological Information System
HL7	Health Level 7
IoT	Internet-of-Things
TDI	Time Delay Integration
HVL	Half Value Layer
KVP	Kilo Volt Peak
HU	Heat Unit
mAs	milli-Ampere Second
PSF	Point Spread Function
FWHM	Full Width at Half Maximum
PD	Poisson Distribution
SNR	Signal to Noise Ratio
ATLS	Advanced Trauma Life Support
AEC	Automatic Exposure Control

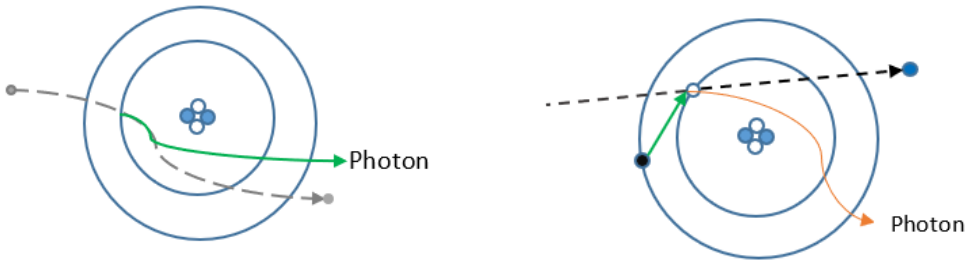
1. INTRODUCTION

X-rays are discovered in 1895 by Dr. Roentgen [1]. During the next century after the discovery, advances in electronics and computer technologies pave the way for invention of new X-ray devices that are able to acquire high quality functional and structural images. These images became and are still valuable in industrial, medical, and security applications. Specifically, the medical X-ray devices (modalities) are categorized into projectional devices which produce 2D images, or 3D scanners (e.g. CT-scanners). Each of these categories grown up into a versatile echo-system of innovative devices which opens new doors in medicine and various other fields. This thesis introduces a unified software solution for projectional devices. Despite the differences in applications, all projectional modalities apply the same physics: generate the X-ray beam, direct it toward the target and turn the at appropriately placed X-ray sensor into an image. This section starts with a survey of X-ray physics, then key components of a projectional scanner is covered in more detail, and known issues with available solutions are discussed at the end.

1.1 Physics of Radiation

When an atom is bombarded with a high energy charged particle two types of phenomenon may happen that end up in X-ray radiation [2]:

- The incoming particle experiences a rapid change in direction, velocity or both. In this case a fraction of its motion energy is transferred into radiation energy. This type of radiation is named *Bremsstrahlung* radiation (Figure 1.1 a).
- The particle may take apart an electron from the inner orbits. In this case, the atom gets into excited state and naturally wants to return back to its steady state. This swinging back to steady state happens when an electron from higher (outer) orbits emits energy in the form of X-ray radiation and fills the empty space. This type of radiation is known as *characteristic* radiation (Figure 1.1 b).



(a) Bremsstrahlung radiation. (b) Characteristic radiation.

Figure 1.1 Different kinds of interactions could emit X-ray radiation: a) Bremsstrahlung radiation is a result of photon distract from its normal pathway. b) characteristic radiation is due to grabbing an electron by an incident photon. The produced gap is filled by another electron from higher energy levels which may release some X-ray photons to settle down in an inner orbit.

The emitted photons carry energy. Relation between photon energy and its frequency is given by [2]:

$$E = hf \tag{1.1}$$

in which photon energy (E) is linearly proportional to it frequency (f). The constant (h) is called Planck's constant (m^2Kg/s).

The *rate* of energy flow is easier to measure rather than counting the photons. This quantity is called *photon intensity* and pictorially shown at Figure 1.2.

Conceptually, radiations are either *monochromatic* in which all photons have the same energy level, or *polychromatic* beam in which a range of photons at different energy participate in the beam. Monochromatic beams are more convenient

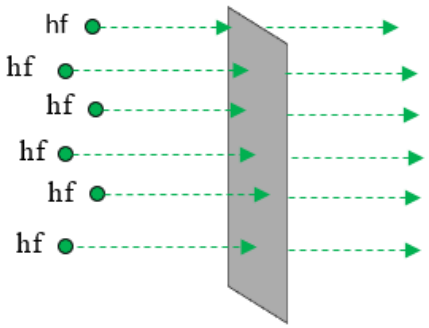


Figure 1.2 Photon intensity is equal to total photons energy per unit area. In a fixed area, increasing photons' energy or its count would increase the intensity.

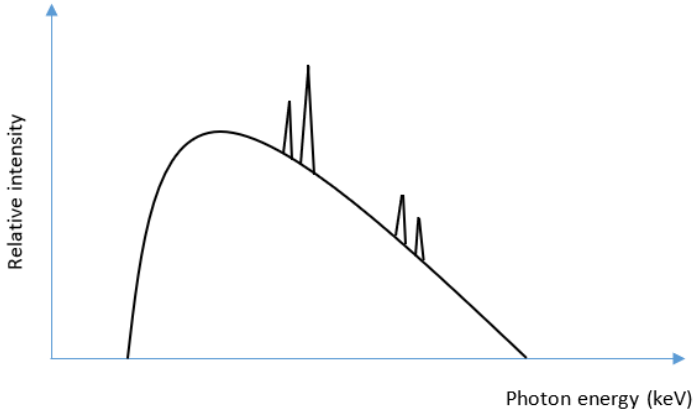


Figure 1.3 Spectral of X-ray production at source. The distribution of this graph depends on the material. Specifically, the sharp point represents the energy levels in which the characteristic emission happens. Their location and height is specific to each material.

for understanding the physics of photons, but in reality, the X-ray sources produce polychromatic beam. For example, bremsstrahlung and characteristic radiation emit spectra of photons at different energies and directions. Also, the photons' intensity at all energy levels are not equal. A pictorial distribution of this intensity is shown in Figure 1.3. The tip points on 1.3 are related to the energy points that the characteristic emission is dominant. The *location* and *intensity* of these points are unique to each atom-type (due to this fact, it is called *characteristic* emission). The curve line shows the bremsstrahlung distribution of the X-ray. The maximum photon energy is achieved when all the kinetic energy of the particle turns into X-ray photon and the lowest photon level is released when the particle has just enough energy to produce X-ray photons.

The monochromatic beam intensity is given [4] by:

$$I\left(hf\right) = \left(\frac{\Delta n.hf}{A}\right)/\Delta t \tag{1.2}$$

In which, Δn is equal to the number of photons with energy hf and A is the area under exposure and Δt is the unit of time. Another physically important fact is that the photon intensity dissipation follows the inverse square low. Suppose at the source of the emission the value of intensity is I_0. At any distance r from the source, the value

of the intensity I is equal to:

$$I \propto \frac{1}{r^2} I_0 \tag{1.3}$$

For polychromatic beam, it is not easy to measure the radiation intensity. So, instead the radiation *exposure* is measured [4]. X-ray is ionizing radiation- it ionizes (i.e. breaks the atoms into charged ion's) the media which it passes through. Consequently, instead of directly counting the number of X-ray photons, the other option is to measure the amount of charged ions which are produced by X-ray radiation. The common unit of measurement is *Roentgen* (R); nonetheless, International System of Units (SI) introduced Gray (Gy) and centigray (cGy) as an alternative. One Roentgen (R), is equal to the amount of radiation which is required to produce $2.58 * 1^{-4}$ coulomb in one kilogram air [4]. Exposure is easy to measure but it does not reflect the whole picture. In medical applications, the amount of produced radiation energy is important factor, but, the questions; such as, what *type* of photon (one should remember that X-ray is not the only ionizing beam and there are other types of ionizer photons do exists) carries the energy, how *many* photons participated in energy transfer beam and how much *time* does the energy transfer take, are also required to be answered when measuring radiation. For example, it is obvious that a set of 100 X-ray photons cumulatively owns more energy than a set of ten X-ray photons. From this example one can conclude that energy is directly depends on the photons' count. As another point, we know that one α particle (another type of ionizer photon) has more energy than even 100 X-ray photons. So, a beam of α particles may be more energetic than a beam of X-ray photons, even though the count of the photons in X-ray is more (unless the X-ray photons count is far more than the α particles). As another hypothetical examination, assume that there is an X-ray beam that produce exactly one photon at each second. In time interval $[0\,t]$ the total number of photons would be equal to t. On the other hand, assume there is another X-ray beam that produces exactly t photons in one second. So, the total transferred energy for both beams is equal. From this last example, one can conclude that energy transfer also depends on the duration of the measurement. All these are considered in a quantity named radiation *dose*. Depends on the applications, there are

different technical definitions for dose [4] such as *absorbed dose*, *exposure dose*, *dose equivalent*, etc. The way these doses are formulated are different but one empirical dose formula for projectional medical X-ray devices is [2]:

$$K_{air}\,(cGy) = 11 * \left(\frac{2.5}{Al}\right) * \left(\frac{KVP}{100}\right)^2 * \left(\frac{A.S}{\left(\frac{r}{100}\right)^2}\right) \tag{1.4}$$

in which Al stands for total aluminum filter thickness (discussed below) on the beam path, r is the distance between beam source and target point in centimeter, s is time (in seconds) that beam is generated and KVP stands for kilo volt peak of the electricity (discussed in the following section). As one can see, the inverse square low also applies for X-ray dose.

X-ray photon has different destiny after hitting an obstacle. Three of them are more interesting in X-ray imaging. First, an incoming X-ray photon directly passes through and does not collide with any atoms. The other is that the photon may collide with an atom on its way and transfers all its energy to that atom and take an electron away. In this scenario, no side photons will be emitted (Figure 1.1 b). This deprecated electron, usually, passes a distance inside the mass and then stops. This interaction is called *photoelectric* and the produced electron is called *photoelectron*. In medical imaging this interaction does not contribute to image formation and just increase patient dose. The other type of interaction is called *compton*. In this case the incoming high energy X-ray photon, produce another low energy photon. This newly born photon's direction is not necessarily same as the original incoming one. Technically, the outgoing photon is called *scatter* photon. Scattering may happen in all directions but its probability is not uniform (with regard to the direction). In Figure 1.4 the cloud around the atom represents the probability of producing compton photon in all directions- The larger the cloud radius, the more probable to have a scatter photon in that direction.

In terms of energy, when an ionizer beam enters a material, it deposits a fraction of its energy within that material via compton and photoelectric phenomena. This energy dissipation is named *attenuation* [2,4] and depends on the material and incoming

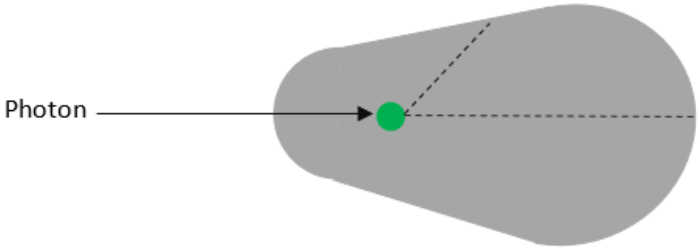


Figure 1.4 Scattering distribution around atom. The more the radius the more the chance of scatter at that direction.

photon's specifications. For example, suppose n X-ray photons enter a material with Δx thickness and Δn photons are absorbed while they are trying to pass through. The relation between incoming and absorbed photons is given by:

$$\Delta n = -\mu.n.\Delta x \tag{1.5}$$

In this relation, μ is called *linear attenuation* coefficient. This coefficient depends on the exposed photon energy, the material density (ρ) and its atomic number (Z). In other words, μ is not a constant for all photon energies. To simplify the relations, usually the linear attenuation is divided by the material density $\frac{\mu}{\rho}$. This quantity ($\frac{\mu}{\rho}$) is called *mass attenuation* coefficient [4]. Similar to linear attenuation coefficient, mass attenuation, also depends on photon energy.

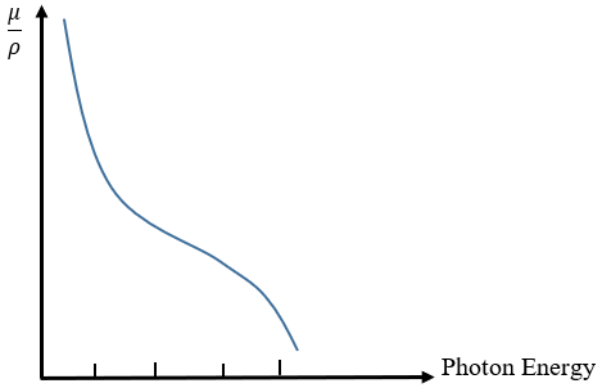


Figure 1.5 Mass attenuation coefficient diagram. To simplify calculations, linear attenuation coefficient is divided by the material density. This new quantity (i.e. material's linear attenuation coefficient over its density) is called mass attenuation coefficient and it depends on the photons energy.

Both linear and mass attenuation coefficients are ideal for monochromatic beam study. To study polychromatic beams, Half Value Layer (HVL) is used. HVL is equal to the required distance in a material in order to reduce the incoming beam intensity to half [2, 4]. The relation between linear attenuation coefficient and HVL is given by:

$$HVL = \frac{0.693}{\mu} \tag{1.6}$$

1.2 Projectional Medical X-ray Devices

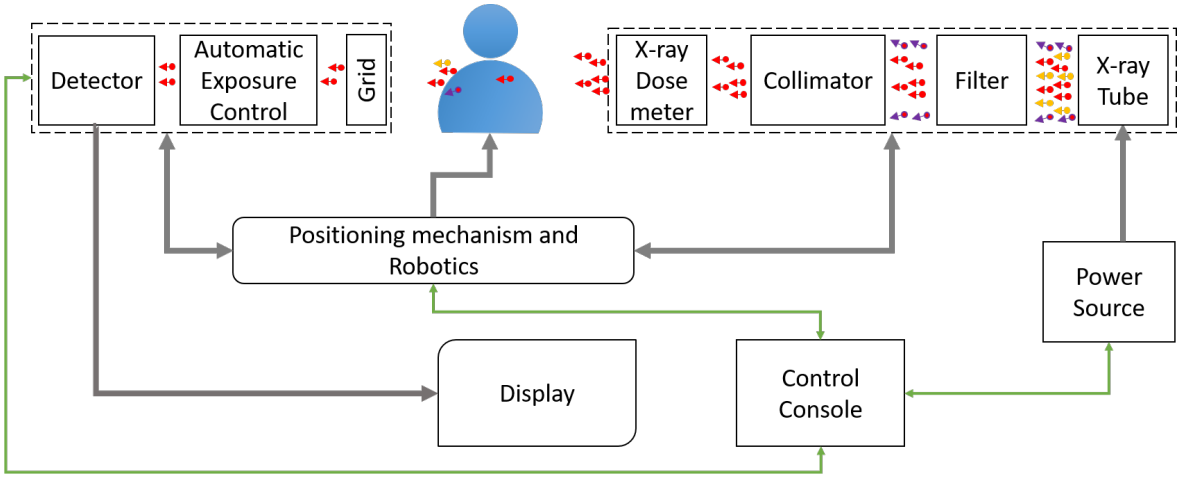


Figure 1.6 Units of a digital radiography scanner. At the production side, a power source produces the electrical power which turns into X-ray beam then the beam passes through filtration and alignment (i.e. collimator) and measurement device. At the construction side, the beam comes out of the patient body is filtered (using a grid), measured for amount of radiation (using the AEC) and then turned into an image.

A projectional radiography device is an assembly of a power source, X-ray tube, collimator, positioning mechanisms, X-ray sensitive receptor and dose control tools (Figure 1.6). The power source generates high voltage electrical energy and conduct it to the X-ray tube. The early model power generators were single phase generators. A simplified schematics of these devices is shown in Figure 1.7. Inside the generator there is a transformer and a rectifier. These components take an AC power input and produce a rail of rectified high voltage peaks. The ripples in the output negatively affect on the produced X-ray. The next generation generators used three-phase electricity. These lines are connected to a special rectifier and transformers to turn into DC voltage. In compare to single phase generators, three-phases generators produces more ripples per

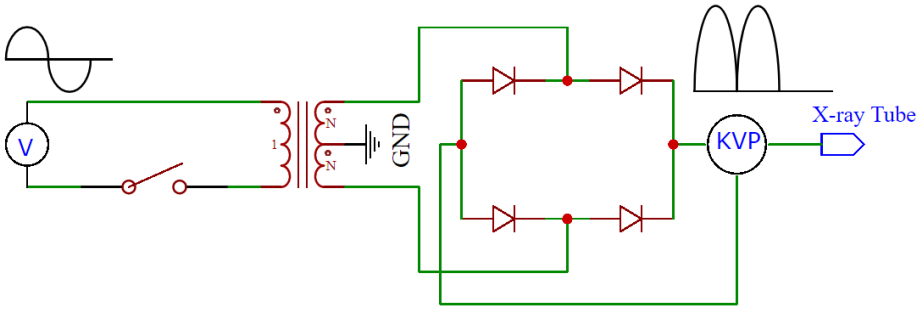


Figure 1.7 Single phase X-ray generator. The output voltage suffers from high ripples and this ripples could damage the device itself and reduce the image quality.

second (e.g. six or twelve), but the magnitude of the ripples are less than the single phase generators. As a result, the three phase power generators were more stable and more efficient than the previous generation [4].

After three-phase generators, medium-frequency generators were introduced. At the heart of these type of generators is an inverter. The inverter produce 5 KHz to 6 KHz sharp DC ripples which easily boost up to thousands of times with special transformer. Next, the transformer output voltage is rectified and smoothed to make a stable output [2].

The major break through in generators technology had happened when micro-controllers were used to control the output voltage [2]. These generators produce high frequency ripples then rectifies them to produce more stable output voltage. Additionally, micro-controllers monitor the amount of heat accumulated on the anode side. Increase in anode temperature beyond the controlled boundaries could damage X-ray tube and negatively affect the output X-ray beam quality. The micro-controllers are pre-programmed to tune the exposure duration such that it stops when the heat reaches the maximum allowed boundaries.

X-ray generation happens within the X-ray tube. Units of an stationary tube is shown in Figure 1.8. Current passes through the filament; electrons *boil off* [1,2] and in the contingency of strong electric field from anode to cathode electrons get accelerated toward anode and hit its surface. Both of the electric field and the filament current

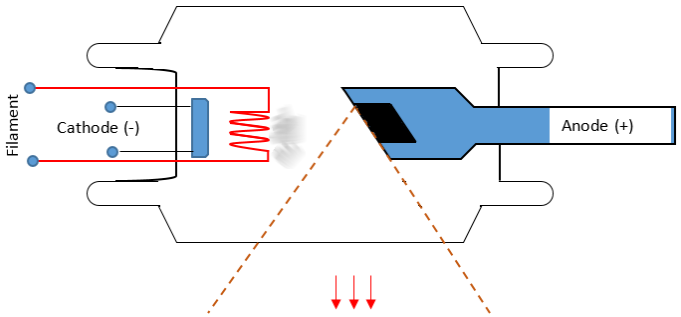


Figure 1.8 X-ray tube internal structure. A filament gets hot and produces a cloud of electrons in the vicinity of a high energy electric field inside the tube. Electrons get accelerated toward the anode pole and collide with the surface and produce X-ray photons.

are produced by a power source. The potential peak difference between anode and the cathode is a couple of tens of Kilo Volt Peak (KVP) and the filament's current is in the range of 3.5 up to 5 amperes [1]. When electrons hit the anode surface, about 99% of their energy is transformed into heat and approximately one percent turns into X-ray [2]. To preserve the produced X-rays, the whole system is compacted inside a vacuum tube. A thin layer of tungsten is used to cover the head of the anode. This is because, the tungsten has high atomic number (Z=74) and the chance of producing X-ray is high. The region on the anode that is covered by tungsten is called *target* [4]. The rest of the anode is made of cooper to easily transfer the heat out of the target area. The region on the target that is bombarded by electrons is called *focal spot* [1, 2]. The head of the anode is tilted with regard to cathode. This angle is called *target angle* [2, 4]. Target angle and focal spot are directly related: reducing the target angle reduces the focal spot size and vice versa.

The intensity of the output beam is not uniform and depends on the direction from focal spot. At the anode site the photons intensity is higher than the cathode side. This phenomenon is called *heel effect* [2]. Figure 1.9 shows the polar diagram of the photon intensity with regard to focal spot. The distance shows the magnitude of intensity at that direction. For example, in this diagram the greatest intensity is achieved at approximately -10 degree tilted toward anode.

By reducing focal spot size, the image gets sharper [8]. However, bombarding on a narrow point on the target would impose much heat stress over the target and reduce

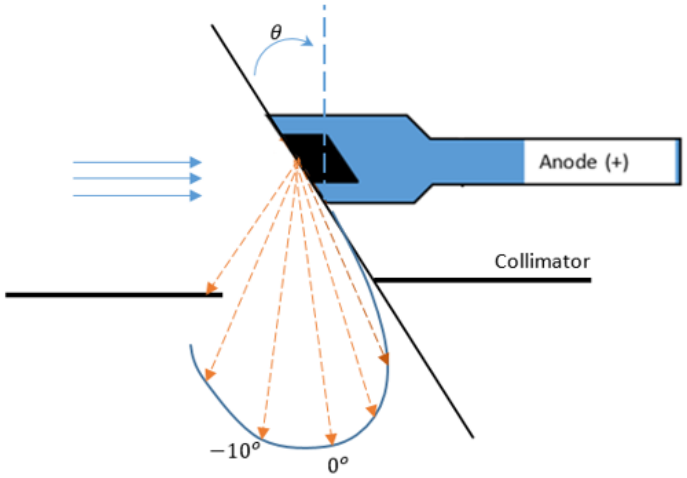


Figure 1.9 Heel effect: the photons are more tilted toward anode pole at the output. To get more effective outputs from device one way is to put the patient on the anode side or filter out the photons that are more close to cathod pole.

the device life time. In 1897 E. Thomson proposed the theory of rotating anodes as a way to increase the anodes thermal capacity. In 1930 the rotating anode X-ray tube has been introduced [1]. The maximum heat tolerance of an X-ray tubes is measured in terms of Heat Unit (HU). Heat Unit is defined as the energy produced by one kilo voltage and one milliampere in one second [2, 4]. For three phase generator it is:

$$HU = kVp * \ mA * \ second * 1.35 \tag{1.7}$$

The X-ray tube and power connectors are contained within a shielded housing. The housing absorbs stray photons (X-ray distribution at anode side is an stochastic and it may distribute in all directions) and provides a reliable high voltage transmission

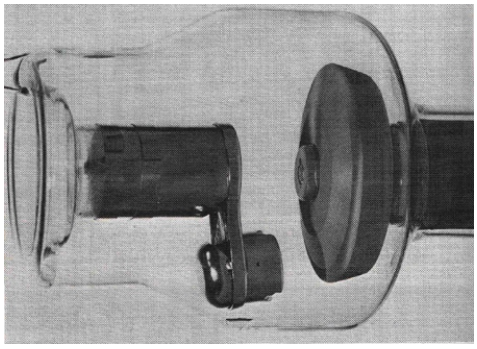


Figure 1.10 Rotational anode was first introduced in 1930 [1]. By rotating the anode, heat is distributed uniformly on the surface. This positively effects the X-ray beam quality device life expansion.

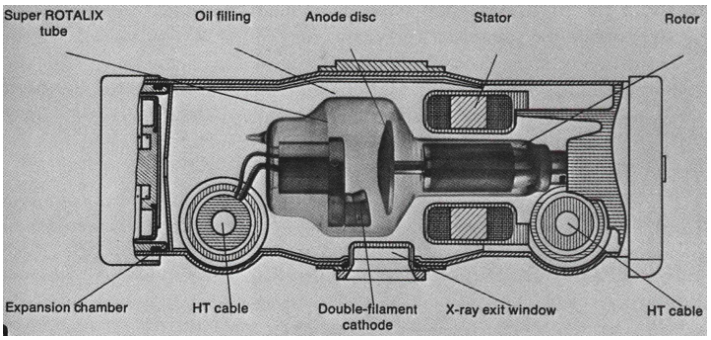


Figure 1.11 Vacuum tube is shielded inside a thick lead housing [1]. In some models, to effectively dissipate the tube heat, the housing is filled with special oil.

lines into the tube. Also, in order to absorb the tube heat it is normally filed with special oil. At the output window of the housing, an aluminum filter is inserted. This filter absorbs low energy photons and only let the high energy ones pass through. The output beam is called *primary beam*. The thickness of this aluminum filter is called *total beam filtration* [4,8] and the process of removing the low energy photons is called *beam hardening* [2]. Even after the filtration the primary beam is not monochromatic. So, the relations that were introduced for attenuation coefficient and HVL is not directly applicable. Alternatively, polychromatic beam is approximated with an equivalent monochromatic beam. For this purpose, special charts (Table 1.1) are prepared that give the HVL of primary beam relative to KVP and total beam filtration [2].

Table 1.1
HVL for different KVP and total beam filtration.

Total beam filtration	Applied KVP			
	60	80	100	120
1.5	-	-	-	-
2.5	2.2	2.7	3.3	4
3.5	2.6	3.2	3.9	4.6

Using the HVL given by the Table 1.1 and locating the corresponding point on Figure 1.12 the approximate photon energy of the primary beam (after filtration) is measurable.

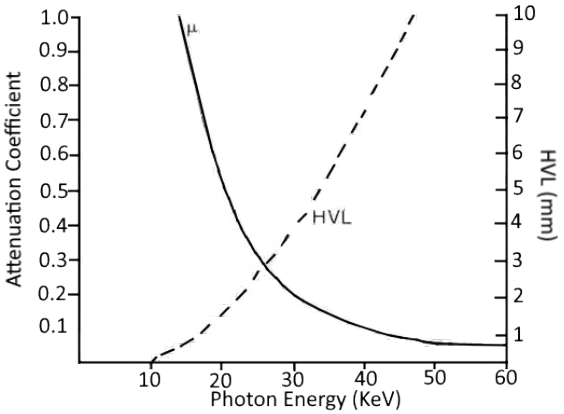


Figure 1.12 Photon energy and equivalent HVL [2]. Relative to less energized photons, higher energy ones could readily pass through materials. Naturally, in order to reduce the intensity of a high energy photons into half, a thicker layer of material is needed.

One of the applications of the equivalent photon energy is to find HVL and/or attenuation coefficient of any tissue. An example of photon energy versus attenuation coefficient diagram is shown at Figure 1.13.

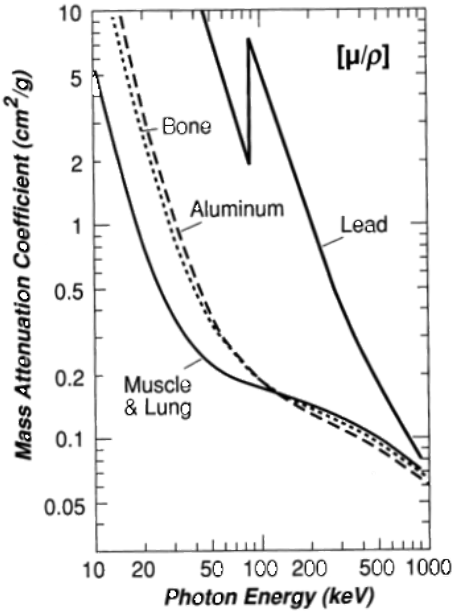


Figure 1.13 Photon energy and attenuation coefficient diagram. Different tissues have different attenuation coefficient at different energies [2]. These differences appears as contrast on X-ray images.

Collimator is used to focus the radiation. The most common collimator type is a rectangle window (Figure 1.14). Inside the collimator, two lead-diaphragms exit: one for adjusting the width and the other for height of the output window. Any photons that hit the diaphragms is absorbed and will not pass.

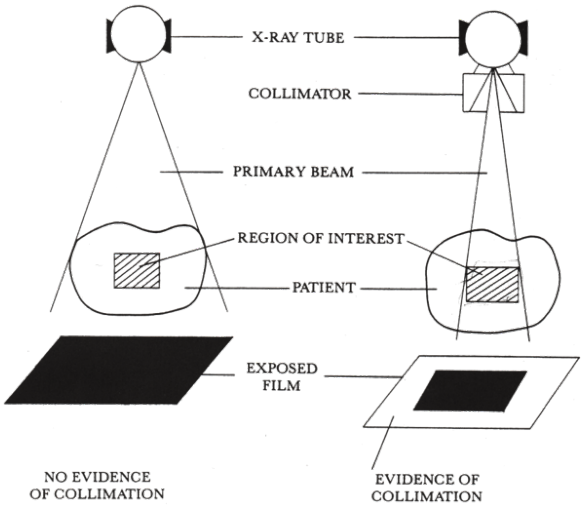


Figure 1.14 In projectional scanners it is common to use a rectangular collimator to focus the output beam on the targeted area [1].

The other common type of collimator is *fan beam* [8]. This collimator produce narrow band X-ray beam (6 mm or less) which is useful for CT-scanners (Figure 1.15).

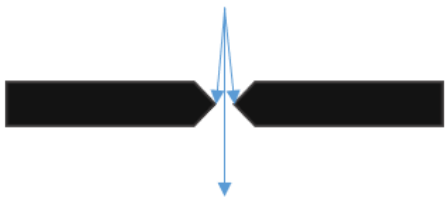


Figure 1.15 A fine X-ray beam can pass through a fan beam collimator and the remaining photons are absorbed.

Photons come out of the collimator, are measured by special dose meter: Dose-Area-Product (DAP) meter (Figure 1.16 a). This device measures the amount of the X-ray dose that passed through a unit of area without affecting the X-ray beam quality. This information is written in the official medical records.

Precise patient-device localization is a necessity for capturing decent images. For this, in some scanners there are mechanisms to position the patient and/or the X-ray source with respect to each other precisely. In early models, these mechanisms were manual or electro-mechanical, but in modern devices they were replaced by robotic arms.

Figure 1.16 DAP meter is used to measure the primary beam dose [3]. This information is used to track the quality of the device.

1.3 Grids and Filtration

The primary beam after passing through an object (e.g. patient) contains two types of beams: *primary beam* and *scatter beam* [4]. The primary beam is the beam that had no interaction while passing through the subject. These photons are informative and positively contribute to image formation. On the other hand, there are scatter photons that have negative effect on the final image. In Figure 1.17, the primary photons directly passed through the body to the other side. The number of photons that successfully passes this path depends on the mass attenuation coefficient and photons' energy. However, at the same point, there are some scatter photons that came from the other paths. These photons cause the point on the receptor to be appear more brightly than what it is in reality.

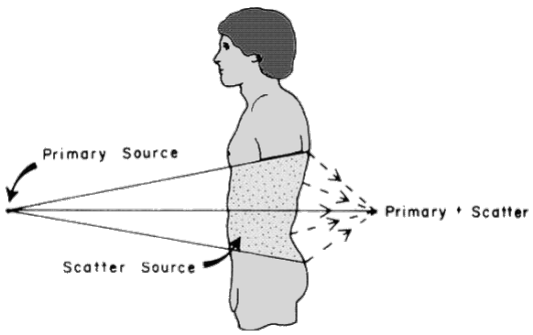
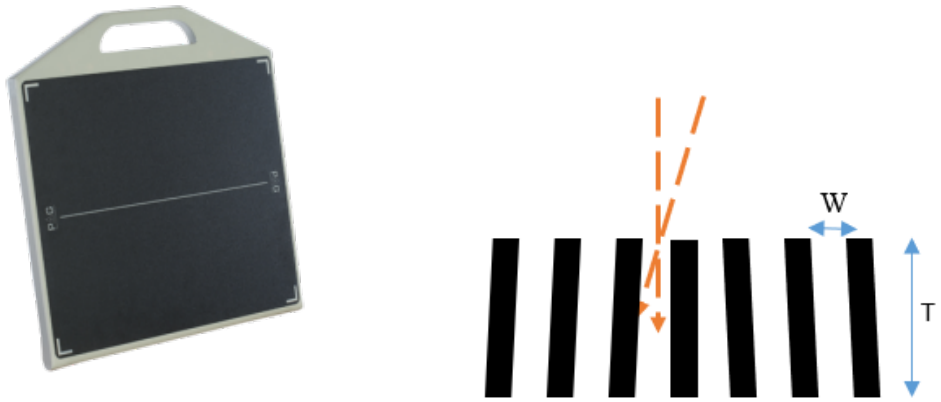


Figure 1.17 Primary and scatter photons come from different paths but ended up at the same location [4]. Primary beam could positively contribute to image construction, but the scatter photons should be eliminated.

To eliminate these photons, two techniques are commonly used. One solution is to insert the image receptor (detector) at a distance such that the scatter photons have minimum chance to get to the receptor. The other one is to use a special apparatus to filter out the scattering photons. This device is called *grid* (Figure 1.18 a). Inside the grids there is a chain of teeth that absorbs the tilted photons (Figure 1.18 B). The



(a) X-ray grid [9]. (b) Grid structure.

Figure 1.18 Grids are used to minimize the effect of scatter photons on the image receptor. The teeth length and the distance in between grids determine the grid ability in removing the scatter photons.

filtration capability of a grid is measured by *grid ratio* and it is defined as [2]:

$$Grid\ Ratio = \frac{T}{W} \tag{1.8}$$

Grids are not perfect filtering devices. They also eliminate a fraction of the primary beam. To compensate this undesirable elimination, *grid factor* should be added on the exposure mAs parameters to compensate the grid lost [2, 4]. This factor, also known as *Bucky factor*, is defined as:

$$Grid\ Ratio = \frac{Incident\ total\ beam}{Transmitted\ total\ beam} \tag{1.9}$$

In modern digital X-ray scanners there is a radio transparent Automatic Exposure Control (AEC) tool. This device measures the amount of dose comes out of the grid without manipulating it. Whenever a sufficient amount of X-ray dose is detected, it automatically cuts the X-ray production.

1.4 Image Formation and Quality

Conventional X-ray images (not CT) are formed from the contrast difference among objects. The mass attenuation coefficient advocate the amount of photons that will be attenuated: the higher is the mass attenuation coefficient, the more photons that were lost while passing through (Figure 1.19).

Technically, the *object contrast* is formulated in different ways [2, 4]:

$$C = \frac{(L_{obj} - L_{back})}{L_{back}} \tag{1.10}$$

In which L_{obj} is intensity of the photons coming from the object andL_{back} is the intensity of photons coming from the background. The other formulations for object contrast are [4]:

$$C = \log_{10}\left(\frac{L_{object}}{L_{back}}\right) \tag{1.11}$$

and

$$C = OD_{back} - OD_{obj} \tag{1.12}$$

In which OD stands for Optical Density and is equal to "the amount of light transmitted through a region of developed film clipped onto a view box" [8].

The mass attenuation coefficient by itself, depends on atomic number of material

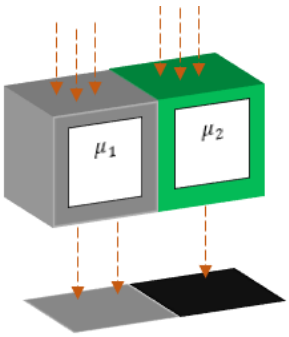


Figure 1.19 Attenuation coefficient differences causes contrast difference on images. Regions with high attenuation coefficient absorbs more photons and appears darker on the image; however, regions with lower attenuation coefficient appears more bright.

under study. In conventional radiography, the mass attenuation coefficient between soft tissues are negligible. Because of this, in angiography and gastrointestinal studies a kind of contrast agent is injected into the patient.

In addition to *object contrast* there are other types of contrast:

- *Receptor contrast* (for film technology): receptors are not equally sensitive. Each model requires a minimum X-ray to sense and can tolerate a maximum amount of exposure [2, 8].
- *Subject contrast*: Refer to the contrast in the pattern of X-ray intensity getting out from the patient and passing through the grid [2, 8].
- *Image contrast*: is determined by both subject contrast and receptor contrast. Image contrast is the product of the receptor contrast by subject contrast [2, 4].

Receptor's *resolution* also an important quality factor. An image receptor resolution represents the measure of details could be reflected on the image grabbed by that receptor. Technically it is defined as "Minimum separation of small objects whose images can just be distinguished from one another" [8]. To measure the resolution, Point Spread Function (PSF) benchmark is commonly used. First, an image of PSF phantom is acquired and then the full-width-at-half-maximum (FWHM) is measured on that image. The inverse of FWHM is equal to the device resolution.

The other image quality factor is *noise*. The photons that hit to the detectors surface follow the Poisson Distribution (PD). Based on PD, if N photons hit to the surface then variance is equal to $\sigma = \sqrt{N}$. In other words, the fluctuation of photons count that hit each region on the receptor is equal to $\sqrt{N}$. In this case the relative distribution is

$$\frac{\sigma}{\mu} = \frac{\sqrt{N}}{N} = \frac{1}{\sqrt{N}} \tag{1.13}$$

and the Signal to Noise Ratio (SNR) is:

$$SNR = \sqrt{N} \tag{1.14}$$

Increasing the number of photons would increase the SNR and consequently, the detectability of the object would increase.

1.5 X-ray Image Receptors

Historically the film technology was the dominant technology in conventional radiology. In this technology, a film is mounted in between one or two intensifying screens [2,4]. These screens are indeed thin sheets of fluorescent materials (Figure 1.20). The film is inserted inside a cassette or a film changer [2, 4]. The intensifying screen absorbs the X-ray photons and turns them into light photons. These light photons pass through the intensifying screen and hit the film. Each point on the intensifier screen that receives more X-ray photons, produces more light photons. These photons, consequently, make the benath region on the film darker (the films are color inverted). The film's emulsion region is much more (100 times) sensitive to light photons than X-ray [2, 4]. So, intensifier (and its quality) plays a key role in films quality. The intensifying screen's performance may vary between 20% and 70%. Factors such as thickness, material and incoming X-ray energy spectrum is important in this regard. The are some difficulties with film technology. First, film maintenance is hard as they are sensitive to environmental factors such as heat, temperature, humidity. Most of the time they were stored in special compartments but still the images faded out by

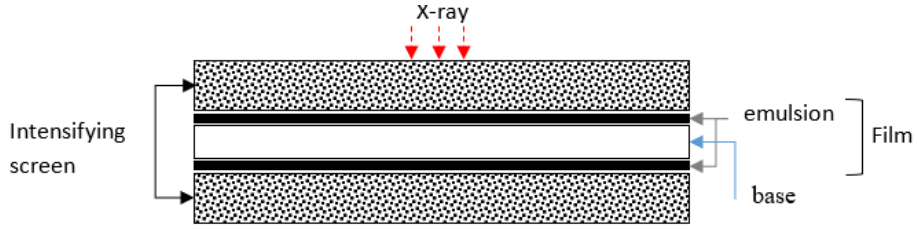


Figure 1.20 Radiology films' base is sandwiched in between intensifiers. These layers absorb X-ray photons on both sides and turn them into light photons. The image quality and its formation are strongly depends on the material and thickness of intensifiers.

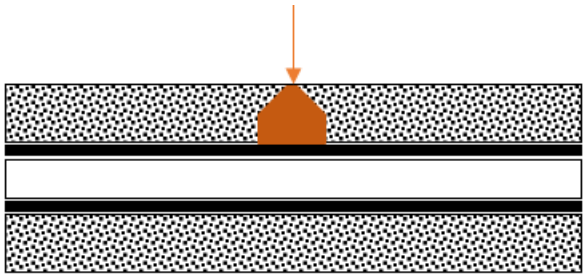


Figure 1.21 X-ray photon produces an avalanche of light photons which appears as an spot on the film. The films resolution determined by this spot size.

the time. The other problem with radiology film is blurring. As shown in Figure 1.21, when the intensifier screen absorbs an X-ray photon it produces a bundle of diverging light photons. This bundle appears as a bigger than what it supposed to be.

Digital radiography generates images in the form of a stream of digits. Early digital scanners digitized the acquired films and then process using a computer. These devices solved the films maintenance problem to some extend; however, the image quality is still limited by the film. The major breakthrough happened after the digital detectors were introduced. In these scanners, images are directly acquired in digital format [5]. Inside the digital detector there is a matrix of detector element (Figure 1.22). Whenever X-ray photons hit these elements, a chain of reactions starts which at the end causes the active area to produce an electric charge. This charge is stored inside a capacitor and during the read-out period, these charges are amplified and processed to generate the pixel intensity at each block.

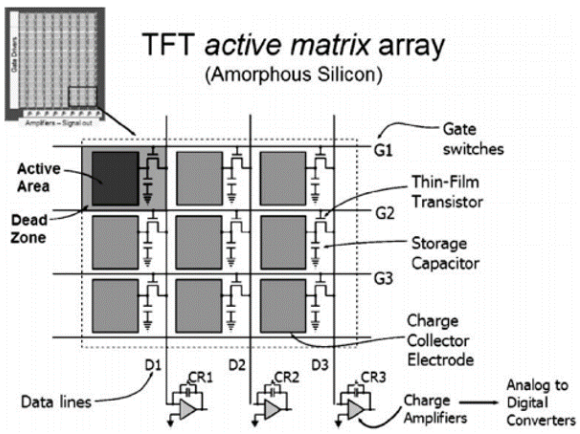


Figure 1.22 Simplified structure of a digital detector [5]. A matrix of active blocks turns the X-ray beam into electrical charges which are pumped into amplifiers and then turned into digital signal. The remaining electronic circuit is for adjusting the timing and speed issues of the active blocks.

From the technology used in the digital detectors, they are either *indirect* or *direct* [5,10]. As it is shown in Figure 1.23, the indirect detectors convert X-ray photons into light photons and then light photons stimulate the charge collecting electrodes.

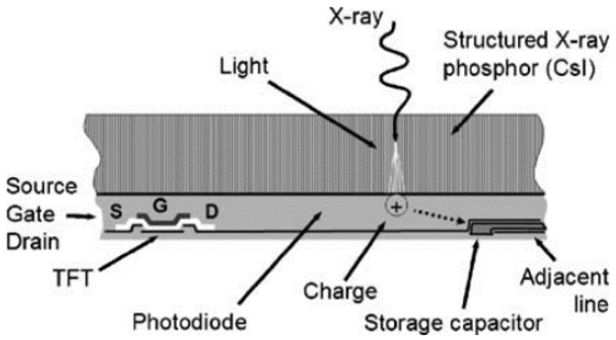


Figure 1.23 Indirect detectors, first convert the X-ray into light photons and then digitize the light photons [5].

The direct detectors are energized from X-ray photons directly and no photon transformation happens inside the detector (Figure 1.24).

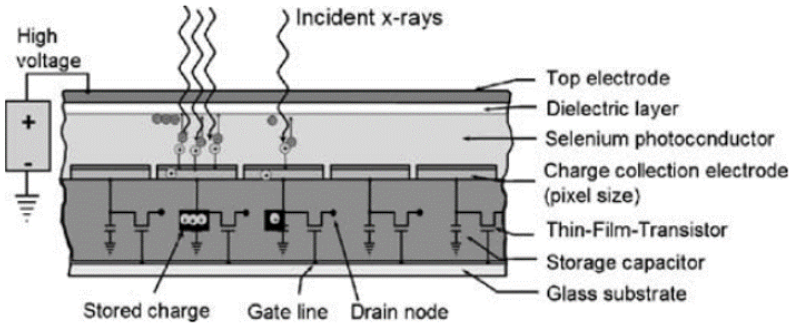


Figure 1.24 Direct detectors do not convert the X-ray photons into light photons [5], but directly turns the X-ray photons into digital image.

In compare to films, flat panels image are easier to prepare, store and more durable. Nevertheless, flat panels still can not solve some problems, such as:

- Both flat panels and films are intrinsically suffer from geometrical magnification (Figure 1.25).
- Blurring and penumbra at the tip points and round objects (Figure 1.26).

These problems are targeted in thin detector design. Instead of a wide flat panels (Figure 1.27 a), a thin detector with Time Delay Integration (TDI) technology

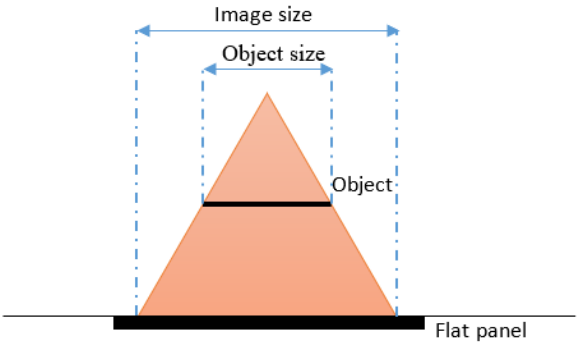


Figure 1.25 Geometrical magnification is intrinsic to digital flat panels and film. Image receptors are inserted inside a Bucky (or under a table) behind the patient. So, there is always a gap between patient body and image receptors.

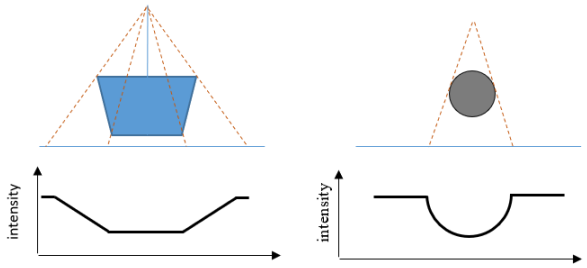


Figure 1.26 Blurring and penumbra reduce the image quality of wide pan detectors. These artifacts depends on the object itself.

is introduced (Figure 1.27 b). In this design the thin detector moves around the target object and scans it line by line.

(a) Flat panel [11] (b) TDI detector [12]

Figure 1.27 Flat panels are produced at different sizes. In medical applications 43 inches is the most commonly used. TDI detectors has different length but their width is less than 2 cm.

TDI scanners are made of multiple lines (in some designs up to 96 [13]) of receptors (Figure 1.28). When device starts imaging, all lines read zero. In the presence of an X-ray radiation the first line reads the X-ray intensity it *senses*. At this moment the other lines still read nothing because they are not exposed to X-ray. The first line records the value that it sense as electric charge in its internal capacitor. By moving the

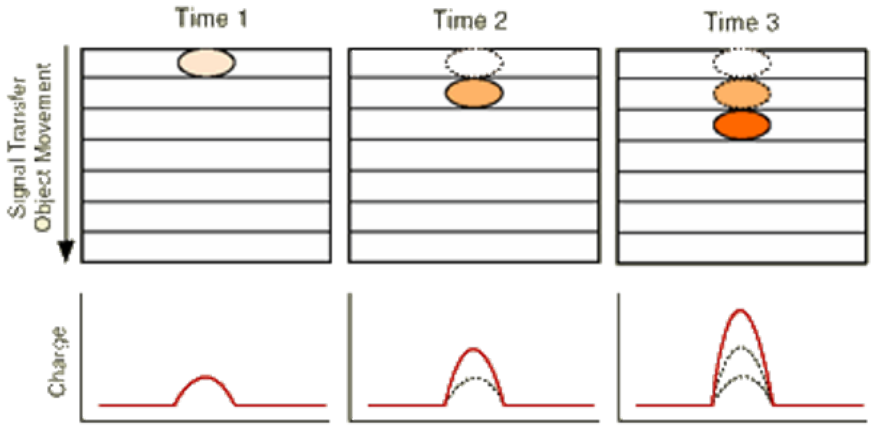
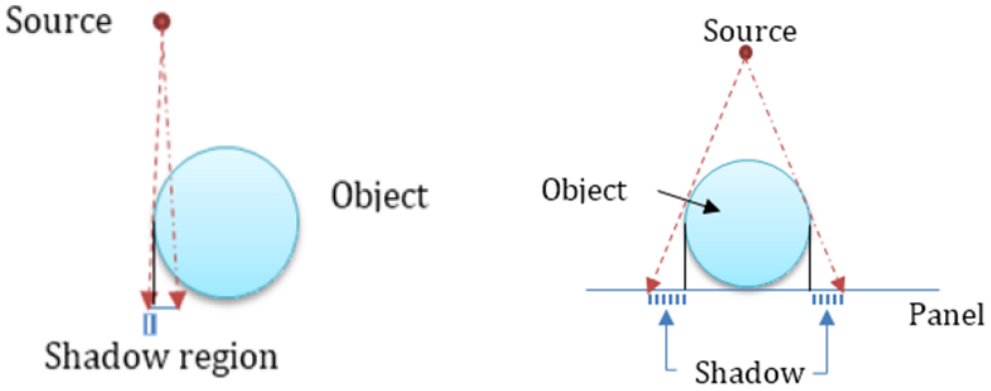


Figure 1.28 TDI detectors construct images step by step and in accumulative way. At each time stamp a fraction of energy is collected and deposited. The output of the last step is integration of energies collected in all previous time stamps.

detector the next step begins. In this step, second line reads the X-ray intensity that it sense and adds it up to the value which already read by the first line. This process continues until all steps have been passed. At the end, the final line produce the final image intensity. It worth to emphasize that the object movement is synchronized with detector line scan speed and at each step, in compare to flat panels, a lower amount of dose is applied.

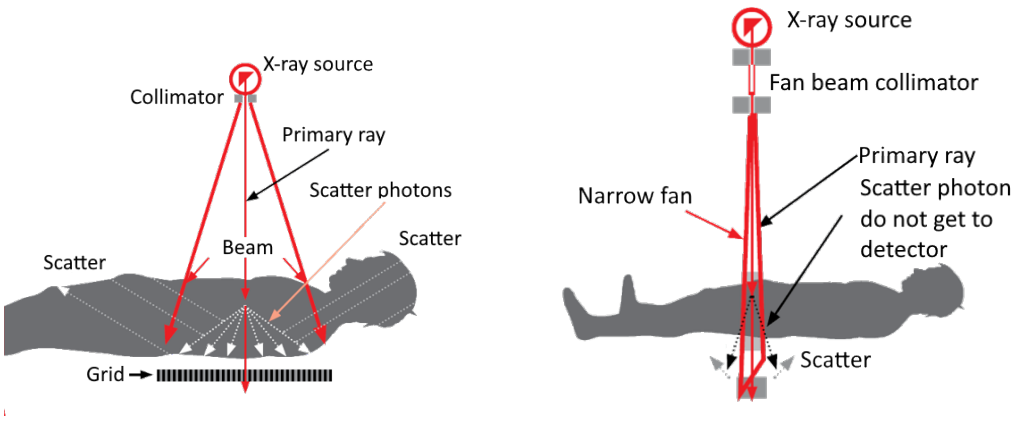
Geometrical image deformation in TDI scanners is less than the flat panels. The slim design of TDI detectors reduces the image magnification and image blurring effects (Figure 1.29 a and Figure 1.29 b).



(a) Object shadow on TDI detector. (b) Object shadow on flat panel.

Figure 1.29 TDI detectors scan at a narrower band and image blurring and object shadow on TDI detectors is less than flat panels.

TDI scanners are less affected by scatter photons. As it is shown in Figure 1.30 a the scatter photons have more chance to hit the flat panel surface than the TDI scanners (Figure 1.30 B). In real world, TDI scanners do not use grid and this reduces the patient dose considerably (because there is no need to compensate the grids factor) [10].



(a) Scatter on flat panel.

(b) Scatter on TDI detector.

Figure 1.30 In compare to flat panels, TDI detectors receive scatter photons in one direction. Because of this, normally, grids are not used within TDI scanners [6].

TDI scanners are more efficiently in using the received X-ray dose. It has been proved that TDI scanners break the inverse square law (Eq. 1.3) and they can make images out of less X-ray dose:

$$I = \frac{1}{r^2} I_0 \tag{1.15}$$

Finally, TDI detectors are more robust against quantum mottling noise [14–16]. To prove this, assume that each step is affected by an independent Poisson noise. In each step, the data value is D_i includes a fraction of the signal S_i and noise n_i:

$$D_i = S_i + n_i \tag{1.16}$$

The output of the the last layer, is the summation of the signals and noises from the previous steps:

$$D = \Sigma_i S_i + \Sigma_i n_i \tag{1.17}$$

in this relation, the $\Sigma_i n_i$ is another Poisson distribution with $\lambda = n\hat{\lambda}$, where n is the

number of stages inside the detector. For TDI scanners the SNR is:

$$SNR = \frac{\mu}{\sigma} = \frac{n\hat{\lambda}}{\sqrt{n}\hat{\lambda}} = \sqrt{n}\hat{\lambda} \tag{1.18}$$

in this relation the SNR proportionally depends on the number of lines within the detector. However, by increasing the layer numbers the sensitivity of the device to vibration increases and it would compromise the SNR advantage.

1.6 X-ray Full Body Imaging

Full body or multiple body part examinations are medically valuable and has a long history in medicine (Figure 1.31). For example, Advanced Trauma Life Support (ATLS) recommends single plain radiography of the chest and pelvis as a part of the primary survey [17–19]. In other cases such as patients who are diagnosed with cancer, multiple myeloma, or other malignancies, there is a need for studies of the entire skeleton to detect the presence or absence of metastatic disease or multifocal involvement [20]. Classically, for such studies isolated images from different body parts are obtained by bedside scanner and then, the images are stitched together to provide a complete picture (Figure 1.32) [14]. In average, image acquisition and stitching takes between eight minutes to 45 minutes to complete [20].

The scan time could be improved by more sophisticated devices that speed up the full body imaging process and reduce the chance of tilted-image-artifact. The first attempts in this direction return back to 1953 [1]. A film changer is used to rapidly change the X-ray films and a timing program (stored on a punch card) accomplish this task(Figure 1.33).

By considering the advantages of the TDI scanners, it seems that they are the best of-the-shelf choice for full body scanning.

Figure 1.31 Judith Allen with her examination record 1930 [7]. Historically, full body imaging is done for bone deformation examinations.

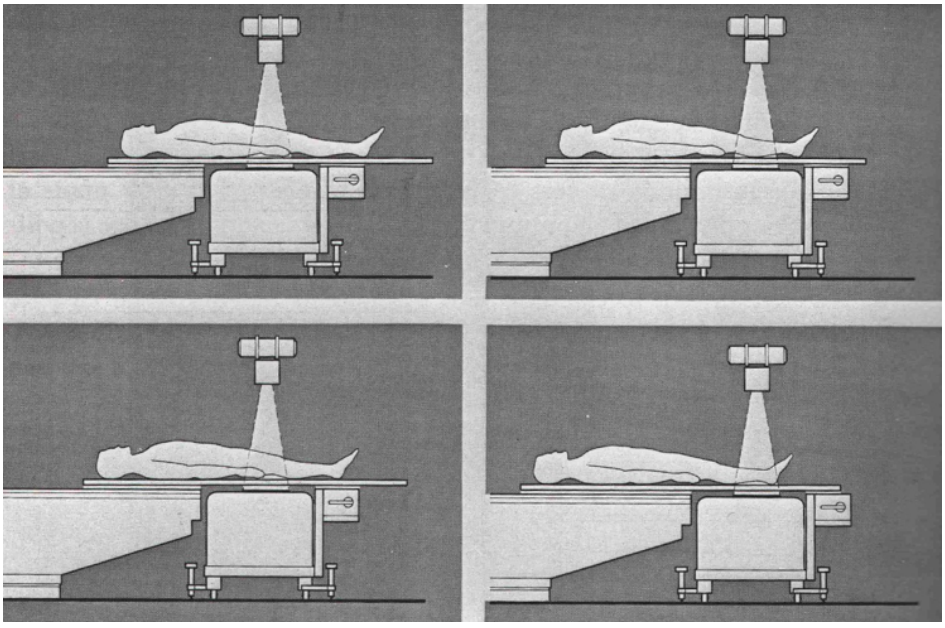


Figure 1.32 First generation full body scanners stitch multiple images together to construct a whole one [1].

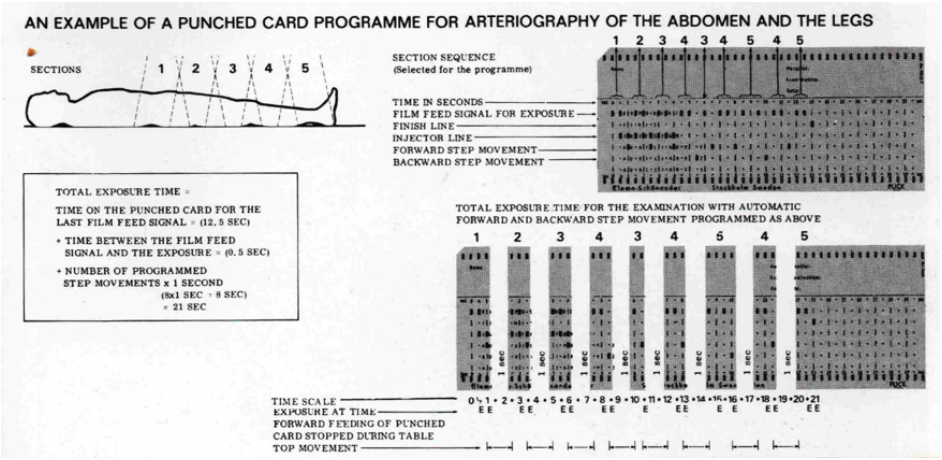


Figure 1.33 After introducing of computers, a film changer and a pre-programmed punch card is used to accomplish a full body examination [1].

1.7 Medical Software

Medical software is defined as: "computer programs, procedures, rules, documentation, and data which is intended for use in the diagnosis, cure, mitigation, treatment, or prevention of disease or other conditions in humans; or is intended to affect the structure or any function of the body of humans; or is used in the production or control of drugs, biologicals, devices, data or other articles produced for use in medical diagnosis or treatment" [21]. The role of software in medicine has increased during the past few decades. Specially, with increasing the computational power of the devices, proliferation of the Internet-of-Things (IoT) in health systems, and advances in machine learning, modern medical devices are more smarter and more interconnected.

Medical society is tightly regulated and standardized. For example, radiology information departments are legislated under: radiological information system (RIS), Health Level 7 (HL7), Digital Imaging and Communications in Medicine (DICOM), and Picture Archiving and Communication System (PACS) standards. The aim of all these standards is to provide a *effective* and *reliable* health service to the patients. For this reason, all non-trivial medical software follows engineered development methods to get these. Still, problems in medical device software result largely from failure to apply engineering techniques, especially during specification of requirements and analysis of human factors [13]. Development teams are often small and have little or no working

knowledge of software engineering methodologies [22]. As a result, observers in the health care field find that it is exceptional to see programs in which design or coding meets good software engineering criteria [23–26]. On the other hand, the complete implementation and application of medical standards in projects require a massive amount of effort, budget, and time. Additionally, the changes and updates of the standards and regulations makes the scene harder for new comers. These challenges are smoothed to some extend by open-source movement. Successful open source projects in the realm of biomedical sciences could be categorized into:

- communication: for example, open source or free HIS/RIS servers, DICOM tools, free PACS servers,
- visualization: for example, VTK, 3D-Sclicer,
- simulation: for example, XRaySim, PENELOPE and PenEasy,
- computer-aided diagnosis: CAD tools,
- image processing : ITK.
- training: OpenClinica.

However there are many realms in medicine that open source is not actively entered yet. One of the aims of this thesis is to fill this gap for biomedical radiography scanners.

1.8 Summary

X-ray beam quality directly affects on the captured radiology image. For medical uses, photons energy is determined by applied KVP. Photon intensity is tuned with the exposure duration(S) and the current(mA). Before introduction of digital-electronics these parameters were used to be tuned with electro-mechanical knobs and buttons. After digital-electronics revolution these devices were replaced by computers and software tools.

The objective of this thesis is to introduce an open source radiography software which is called iBEX. This package satisfies basic requirements of a digital radiography machine such as tuning X-ray parameters. Additionally, the software controls and monitors the device components status and log events, fetches the new imaging tasks list from RIS. Also, the captured images are viewed, processed, and stored on local storage and sometimes later are sent to PACS server. All communications and transactions are compatible with DICOM standard.

iBEX is developed for a TDI based full body scanner. Its assessments were accomplished on this device. However, its modular design is integrable with other components and installed on other machines. Additionally, researchers could develop their own filtering plug-ins and integrate them into iBEX. These extension mechanisms are unique on its own kind and there is no counterpart available in the readily for researchers in this field.

In the following, chapter two discusses the iBEX itself in detail and in chapter three a framework in which iBEX is developed is discussed. This framework could be used to extend iBEX itself or similar projects which require to follow up medical standards. Finally chapter four concludes the thesis and lists the future works.

2. iBEX: MODULAR OPEN SOURCE SOFTWARE FOR DIGITAL RADIOGRAPHY

A device independent software package, named iBEX, is developed to accelerate the research and development efforts for X-ray imaging setups such as chest radiography, linear and multi-directional tomography, and dental and skeletal radiography. Its extension mechanism makes the software adaptable for a wide range of digital X-ray imaging hardware combinations and provides capabilities for researchers to develop image processing plug-ins. Independent of the X-ray sensor technology, iBEX could integrate with heterogeneous communication channels of digital detectors. iBEX is a freeware option for preclinical and early-clinical-testing of radiography devices. It provides tools to calibrate the device, integrate to health information infrastructure, acquire image, store studies on local storage and send them to Picture Archiving and Communication System (PACS). iBEX is a unique open-source project bringing X-ray imaging devices' software into the scope of the open-source community to reduce the X-ray scanners research effort, potentially increase the image quality and cut down the production and testing costs of radiography devices.

2.1 Introduction

A projectional radiography device is an assembly of a power source, X-ray tube, collimator, positioning mechanisms, X-ray sensitive receptor, and dose control tools. The power source generates a suitable range of electrical energy and conduct it to the X-ray tube to generate an X-ray beam. A collimator at the window of the X-ray tube housing influences the size and shape of the X-ray beam exiting the tube assembly and directed toward the patient body. Within the body, X-ray photons may be attenuated or directly pass through. The latter photons are called primary photons and positively contribute to the image formation at the detector. However, attenuated photons have the chance to find a way out and bring unwanted contributions at the detector. These photons are called scatter photons and are filtered by a specifically designed grid. A

detector captures the received photons and forms an image [4], in which, each pixel intensity is directly related to the amount of the incident X-ray photons. The parameters of early projectional radiography scanners were manual and adjusted by an operator using electro-mechanical knobs and switches, but in time, devices have adopted such that the software's role in automation and control of the processes has increased. In 1970s, High-Frequency (HF) power generators were introduced which outperformed three-phase power generators in terms of efficiency and reliability [2]. In this type of generators, an embedded software in a microcontroller was responsible for generating high frequency switching signals, which were applied for controlling the output voltage. Indeed, it was the first appearance of a software unit within a radiography device. During the 80s, electro-mechanical timers and knobs started to be replaced by digital and electronic counterparts [1]. By the early 90s, a form of an embedded software component was presented in almost all radiography devices. At this decade, digital revolution completely changed the radiology devices, which result in: 1) introduction of digital X-ray detectors, 2) significant reduction in storage costs, and 3) proliferation of digital communication infrastructures in medical centers. These advancements all together create a new ecosystem. The radiography devices need to be upgraded to fit to the new environment. Despite the successful history of the embedded solutions, they were not enough for these new demands. Due to this issue, embedded digital hardware was replaced by general purpose computers and software tools were developed to handle the necessary requirements and controlling tasks. From then, researches are focused on optimizing the digital radiography systems; specifically, with the aim to acquire better images from lower X-ray dose. As a result, receptor technology and material [27, 28], collimator design [29], acquisition protocol guidelines [30] and image processing and filtering [31] were all subject to further optimization. Also, studies were conducted to compare different imaging modalities [32] or image quality of the same modality from different vendors [33] and to update the standards [34]. These studies have highlighted the fact that despite a long history of the projectional radiography, it is still an active field of study with continuously emerging dynamic requirements. However, manufacturers develop their own hardware and accompanied software packages as a closed-system and some vendors limit access to critical sections within their software products. This approach is attributed to the tightly regulated ecosystem of

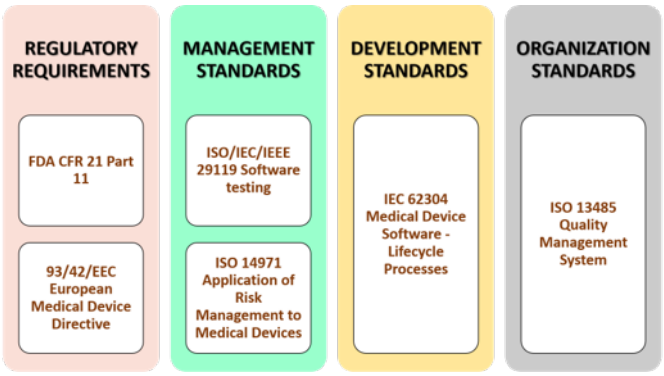


Figure 2.1 FDA and EMA standardized medical software development, management and integration activities. Some of the most commonly applied standards and their corresponding categories are shown.

the medical device environment. Figure 2.1 shows a common set of standards that are related to medical software.

A medical software, either as a part of a device or stand-alone medical application, is required to comply with several regulations, like U.S. Food and Drug Administration (FDA) regulations on electronic records and electronic signatures, CGMP-CFR 21 Part 11 [35], or European Medicines Agency (EMA) 93/42/EEC directives [36]. A set of standards were prepared for software products: IEC 62304 (Medical Device Software – Life Cycle Processes) [37], which prescribes software engineering-related matters, ISO/IEC/IEEE 29119 (software testing) family, which defines software testing standards that is usable in any development methodology; specifically, ISO/IEC/IEEE 29119-5 [38] promotes Keyword Driven Testing (KDT) [39] as an efficient and consistent solution for evaluating software units. IEC 14971 [40] discusses a risk management framework within which the safety-critical properties are hold and guarantees conformance to abstract specification of safe device operation. Additionally, there are standards that are not specific to software alone, but organizations should provide evidences of conformance; like, ISO 13485 (Quality Management System). Companies that successfully pass this pathway for their products, prefer to close their achievements for external intervention and this effectively limits the flexibility for researchers. In opposite to the closed-form manufacturing paradigm, various kinds of dedicated X-ray devices could be brought together, from off-the-shelf components made by different manufacturers. For example, a power source from vendor A and a detector from vendor

B may be used to make a new X-ray scanner device. This assembling paradigm liberate the researchers from prefabricated devices limitations and allow for customization of the radiography device based on their own needs and requirements. The only issue with the new paradigm is the lack of a device independent and general-purpose software package which could interact with the components from different vendors. Such a software package would complete the chain of the customized X-ray device fabrication and its requirements have already been defined by American College of Radiology (ACR), the American Association of Physicists in Medicine (AAPM), and the Society for Imaging Informatics in Medicine (SIIM) [41]. The purpose of this study is to provide a multitude of modular open-source software tools, which is named iBEX, to facilitate the workflow necessary to build a functional digital radiography system. Its plug-in mechanism paves the way for integrating devices from different vendors and developing custom image processing algorithms. In order to reduce the platform compatibility issues, iBEX was written in C++ and it is portable to any state-of-the-art operating system and hardware architecture (i.e. x86 or 64-bit system). Researchers and developers could download the source code from the project GitHub repository [42] and rebuild it for their own scanners. SOLID principles [43] and Model-View-Controller (MVC) design patterns [44] were applied during iBEX development, rendering the output source code neat, safe and efficient. These goals were evaluated by means of automated testing tools and profiling widgets. Furthermore, iBEX functionality and reliability were assessed on an innovative full body Time Delay Integration (TDI) based X-ray scanner. In the following sections; iBEX architecture and major workflows are presented and then its novel extension methods are discussed in detail. In the results section, iBEX analysis results are presented. Finally, at the discussion section, the iBEX, current position, and its future in the medical open source development are discussed.

2.2 Materials and Methods

This section reviews iBEX internal from normal users and technical developers' perspectives. Normal users get benefit of main workflow and high-level functionalities of iBEX solely for imaging and device calibration purposes without dealing with

technical details behind. On the other hand, technical developers access the full iBEX code. These users are normally responsible for developing, installing, updating, configuring, and customizing iBEX for a new imaging device. The project development follows IEC 62304 standard [37]. This standard discusses a generic Application Lifecycle Management (ALM) for medical software and its minimum required criteria, without involving methods and realization techniques. Any methodologies that meet these minimums could be acceptable. Among the available options, Rational Unified Process (RUP) [45] has been selected for iBEX realization. First, the requirement-analysis document of the project was prepared. Based on that document, the project was divided into three milestones. Then, a design document of the first milestone was sketched out. In an iterative and incremental process, the requirements document, design artifacts, and prepared source code were gradually updated. This process continued until all targets of a given milestone were achieved. Then, the next milestones were started and continued till all the requirements were satisfied. In this section, the focus was on the final outputs and the details of each iteration were omitted.

2.2.1 Requirement Analysis

The requirement analysis started from ACR–AAPM–SIIM guideline [41] and Digital Imaging and Communications in Medicine Information Object Definition (DICOM IOD) of digital radiography [46]. Based on these guidelines, a list of required workflows was identified, then, the list was enriched by some of the other workflows that are commonly available in modern digital radiography scanners (Table 2.1). This list was the base of all analysis, implementation and test designs.

2.2.2 System Design

iBEX had a micro-kernel architecture (Figure 2.2). All major workflows (Table 2.1) and a plug-in manager were implemented within the core package.

Table 2.1
Digital radiography common requirements.

Workflow	Description
register new patient	capture the patient's demographic information and record it in the local database.
view image	load local image file (with tiff or DCM extension) and display it on the screen
select imaging task	opens the local task list to select a task.
quick scan	starts to acquire a study without registering the patient. It is useful for emergency scans
configure devices	updates parameters of components (such as detector binning mode or energization level of power source).
apply image filtering	updates the images by applying a filter plug-in (one could also make changes on the filter parameters before its application).
manage local database	checks the database for data integrity and controls the capacity.
decide on image	after acquiring an image, the user can decide to accept or reject the image. If the image was rejected, then the acquisition is expected to be repeated.
log event	Events within the system are recorded in different levels (e.g. warning, information, error, fail).
send to PACS	Finished studies are stored in a remote PACS server. The server connection settings should be configurable.
configure machine	Peripheral components' calibration parameters are updated.
make DICOM file	converts the raw image into a DICOM file.
create account	User credentials are updated.
update worklist	Modality worklist is automatically (and manually) updated.

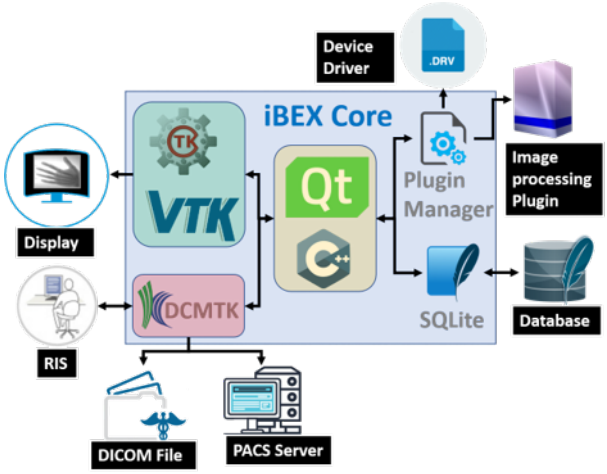


Figure 2.2 iBEX core package is the kernel of the system. It includes all major workflows and a plug-in manager.

Internally, iBEX core contained VTK [47], DCMTK [48], CTK [49], Qt [50], Log4Qt [51] and SQLite [52] packages. Each package had a specific responsibility within the kernel. VTK library was used for image rendering, DCMTK was applied for DICOM transactions and file format conversion, CTK has built-in plug-ins and tools for managing PACS connection and provided extra Qt widgets for handling and viewing images. The plug-in manager was based on Qt plug-in engine and log-engine leverages Log4Qt package. Finally, SQLite was applied for managing database.

2.2.3 Implementation Concerns

iBEX code was developed based on well-known best practices and guidelines. The source code clarity and reusability were boosted by applying separation of concern principle [53] and implementing MVC design pattern. The view objects were responsible only for interacting with the user interface, model objects stored and retrieved data, and controller objects handled the data flow in between view objects and model objects. The events were divided into systemic and application categories. Each category is logged separately. Systemic events include technical details and mostly designed for debugging iBEX itself. The application events observe users' interactions like the last time a user signed in. The log files are stored in a rolling fashion in text format, and when their sizes reach a predefined amount (current limit is 10MB), they

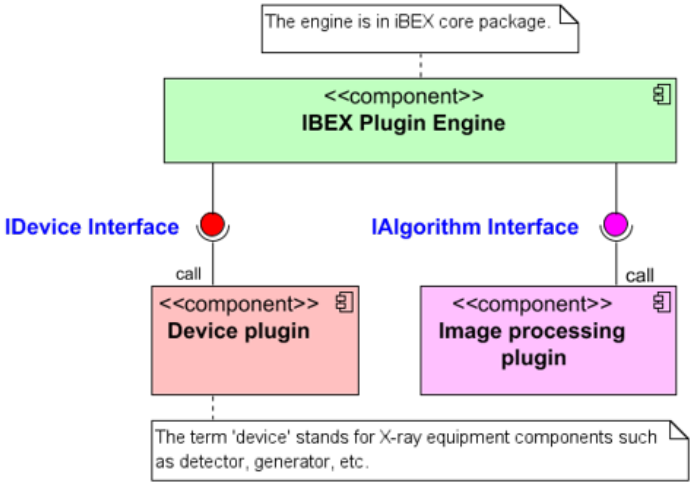


Figure 2.3 iBEX extension mechanisms supports device and algorithm extensions via IDevice and IAlgorithm interfaces.

are overwritten. DICOM standard has a database model [54] for DICOM compatible medical X-ray scanners. In this model, each patient entity accomplishes at least one study entity, each study contains zero or more series entities and each series contains zero or more instances entities (i.e. image). iBEX database implements this model with solidifying two design choices: first, all the patient's imaging tasks are completed in one session. So, each study contains one and only one series. Second, each instance (i.e. image) is independent of the others and stored in a separate file.

2.2.4 Extension Mechanisms

Plug-in manager of iBEX core supports two types of plug-ins: device and algorithm. The device plug-in provides a mechanism to integrate a device driver, for example a detector driver, and algorithm plug-in is for merging new image processing algorithms into iBEX. As it is shown in Figure 2.3, IDevice and IAlgorithm interfaces provide the common language in between iBEX core and external plug-ins. During iBEX core boot period, plug-in engine automatically searches for the files that implement one of these interfaces and loads them as extension plug-ins.

The IDevice interface standardizes the commands and massages that is sent and received in between iBEX core and peripheral devices (Figure 2.4). This interface iden-

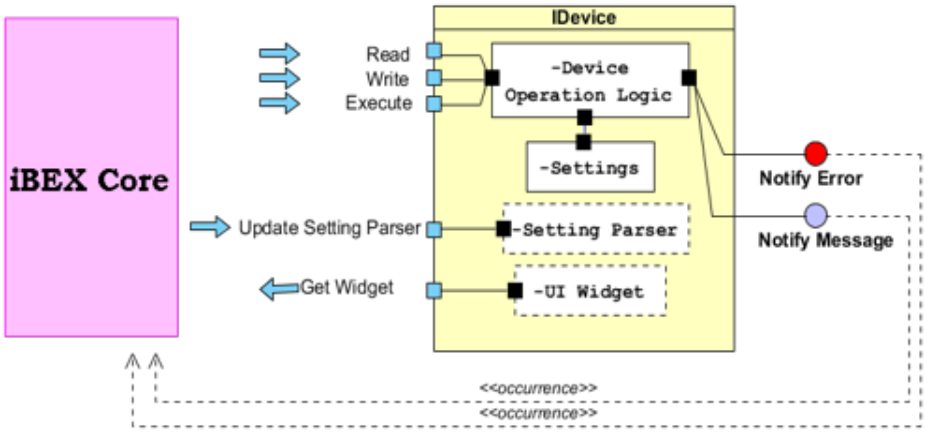


Figure 2.4 IDevice interface. This interface describes a common context in between iBEX Core and a device plug-in.

tifies three types of operations for each device: read, write and execute. To accomplish each of these commands the plug-in may use a setting value. IDevice could discriminate error and normal messages. These notifications are fired from the device plug-ins and contain the situation information. They are captured and handled within iBEX core. Some device plug-ins may have a user interface to display on the screen. iBEX core, by means of the 'Get Widget' function, checks whether there is any user interface widget within the IDevice plug-in or not. If there is any, then it would load and fuse it to its own interface during the boot time. Additionally, IDevice interface discriminates volatile and permanent settings. Volatile settings are parameters restricted to the current session and disappear when the session is ended (or due to power loss). On the other hand, the permanent settings are stored within the device firmware, and they are not cleared even after the session is finished. To handle this latter type of settings, iBEX core requests a parser in order to help with permanent settings. Some permanent settings, like detector gain and offset, could critically affect the device performance. Manufacturers prefer to block the end user access completely or intentionally to make it hard to manipulate those calibration parameters. However, if the user is authorized and provided a way to update the critical calibration parameters, then it is feasible to change them by means of an IDevice plug-in.

The IAlgorithm interface is specialized for image processing (Figure 2.5). iBEX core passes the raw images to an IAlgorithm filter via 'Set Input' command. The

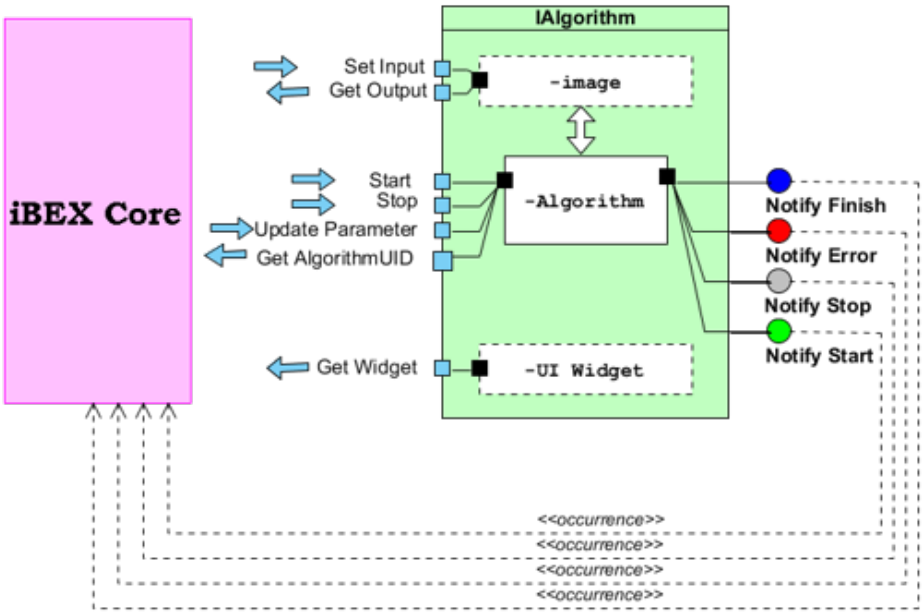


Figure 2.5 IAlgorithm interface. Image, algorithm and UI widget are IAlgorithm properties. iBEX core interacts with a filtering plug-in via a set of functions defined in the interface.

processed image is reloaded back to the iBEX core with 'Get Output'. The core can trigger the algorithmic operation of the plug-in with 'Start' and 'Stop' command and through the 'Update Parameter' informs the IAlgorithm plug-ins with the most updated filtering parameters. The plug-ins could inform the operation status to the iBEX core via firing notification messages (i.e. Finish, Error, Stop and Start) and the iBEX core handles these messages appropriately. Like IDevice, IAlgorithm plug-ins may contain a graphical interface widget and, likewise, the 'Get Widget' function returns this interface from the plug-in into the core.

IDevice and IAlgorithm interfaces intrinsically differ from each other in the way that they handle the commands. It is reasonable to assume that the device commands are executed in relatively shorter time than algorithmic operations, which may take a while to complete. Based on this, IDevice operations (read, write and execute commands) are conducted synchronously, while IAlgorithm operations (start and stop commands) are asynchronous. In other words, whenever iBEX core issues a device operation command to a device-plug-in, it blocks itself and waits until the command is finished, or a notification message is returned. On the other hand, when iBEX core starts (or stops) a filtering task within an IAlgorithm plug-in, it immediately returns

to its own execution thread. It is the plug-in's responsibility to inform the core with appropriate notification messages, such as start, stop, progress, or error notifications.

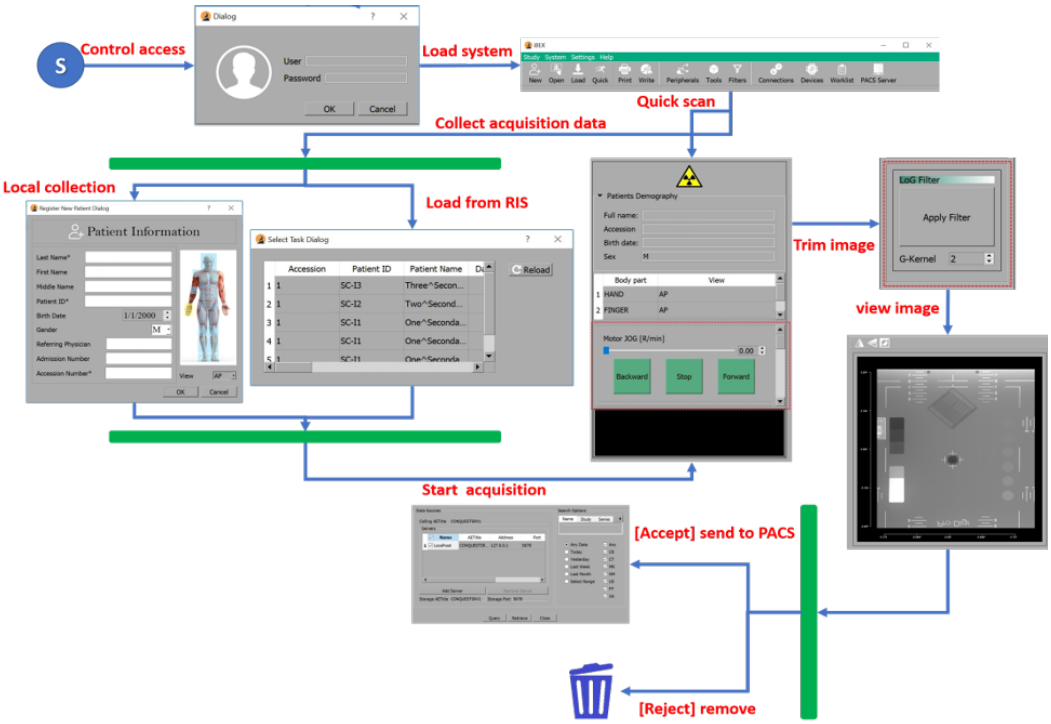


Figure 2.6 iBEX internal main workflow. This workflow is used by normal users. Most modern scanners have three ways to get start their tasks: a) register a patient on the machine b) fetch the patient data from RIS server c) accomplish an examination without registering a patient information. After examination, user can apply IAlgorithm filters on the image. If the result was satisfactory then it would be saved in assigned PACK server, otherwise, it could be dismissed.

2.2.5 Main Workflow

iBEX main workflow is shown in Figure 2.6. After successful login to the system, the main menu shows up. Each item on this menu targets a section of the common workflow (Table 2.1). iBEX provides three methods to start a study:

1. Register a new patient: the patient's DICOM critical demographic information is first collected and registered at the local database.

2. Fetch from Radiology Information System (RIS): imaging tasks and patient demographics are fetched from the RIS system.

3. Quick operation: imaging task is performed with no need to collect demographic details. This is useful for emergency and anonymous studies.

The 'Examination dialog' controls the examination exposure parameters and flows. After successful acquisition, image processing task is initiated via the filters' widget. The processed image is shown in the image viewer section of the 'Examination dialog' and if it is acceptable, then it is sent to PACS server, otherwise it is eliminated. Other workflows such as updating PACS and RIS servers' connection settings, or registering new users are also available but are not shown in Figure 2.6.

2.2.6 Filtering Pipeline

The digital X-ray images are large and require a delicate flow pipeline to prevent memory leakage, or system crash. To optimize the data transfer in between iBEX core and IAlgorithm plug-ins, the raw images were encapsulated in multi-layer data structure (Figure 2.7). First, the raw images were converted into raw 'vtkImageData' type. This data type is consistent with the other image processing and filtering tools of the VTK library. Then, to activate the VTK memory management, the 'vtkImageData' was wrapped with a smart pointer and then enlisted in Qt list data structure. This list is passed to the first filter plug-in. When it is finished with processing, it notifies the core with the filtered image. This flow is repeated for the next filter until all filtering tasks are finished. Then iBEX core fetches the processed image from the last filter and displays the output on the screen.

2.2.7 Tests, Risks and Measures

In harmony with IEC 14971, risk management activity placed underneath of the project management discipline. Risks and their associated harm were studied before any piece of code was written. Then, candidate routines, practices and control mechanisms that could avoid or reduce harm severity were devised and reported on

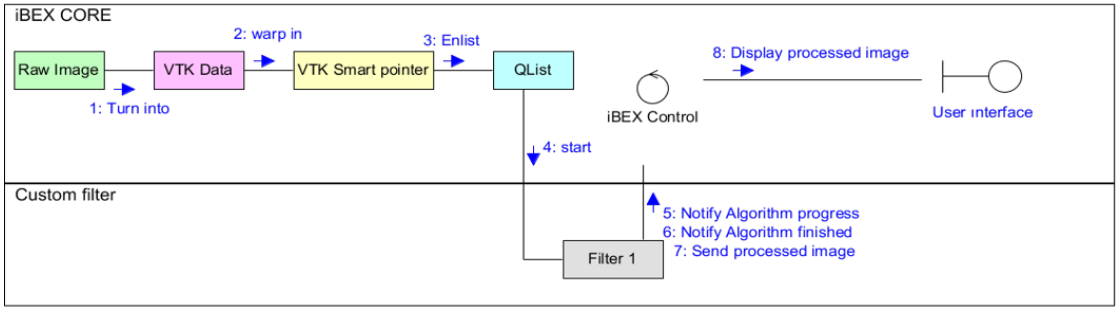


Figure 2.7 Data flow pipeline between core and first custom filters. If another filter exists, then iBEX automatically activates. The final filter output is displayed on the screen.

Failure Mode and Effects Analysis (FMEA) document- as a part of analysis-and-design artifact. By this effort, iBEX would always stay within a tolerable harm zone. In accordance with ISO/IEC/IEEE 29119-5 standard, iBEX tests were accomplished based on two layers KDT model: domain layer and test interface layer. Each domain-layer-keywords associates with a group of tests at interface layer and represent the highest abstraction level. However, test interface layer keywords stand for the most atomic interactions and specify the lowest level of abstraction. After development, the released executable was tested on an innovative full-body TDI X-ray scanner that we had developed (Figure 2.8). In this device, one Gulmay CF-series power generator through a serial port, a Delta ASD-b2 motor driver through a MODBUS RTU channel and one Teledyne Argus TDI detector through an ethernet channel were connected to the workstation computer. Additionally, on the workstation there were a PACS server [55] and a RIS emulator [56] to simulate real environment. This case study is used to profile iBEX runtime resource management and measure the code coverage [57]. To best of our knowledge, no medical software standards suggest rigid boundaries, methods, or criteria for quantitatively measure the medical software quality. However, commonly used criteria such as count of activated functions [58], number of passed conditions [58], amount of used memory, CPU peak and GPU load were measured for iBEX.

2.3 Results

Using the common requirements (Table 2.1), iBEX was logically organized into a set of domains and for each domain a keyword was assigned. Respectively, each

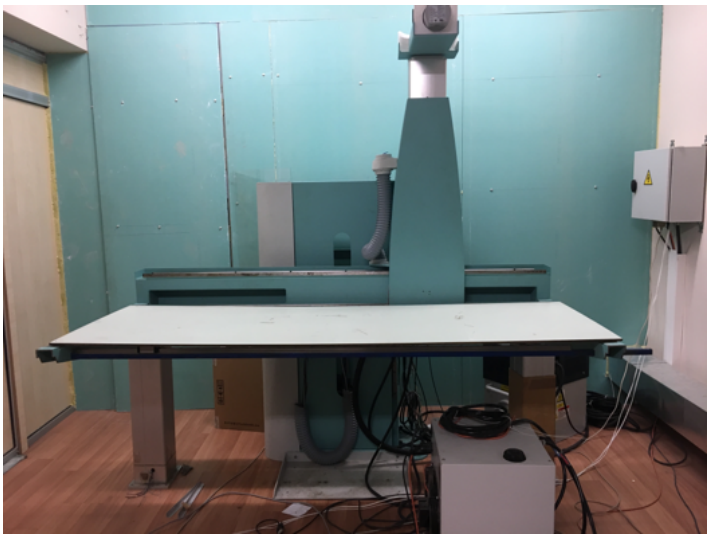


Figure 2.8 Developed full body X-ray scanner. In this device a TDI detector, high frequency power generator and three servo motor drivers are required to be controlled with iBEX.

domain is divided into atomic operation units and tested individually. Some tests are accomplished by automated testing tool, marked as AUTO, and the others are done manually, labeled MAN. Except a few warning messages, majority of tests passed successfully. The reason of the warnings was in the fact that the VTK is in transition from OpenGL to openGL2 backend technology and there is inconsistency in between VTK and CTK widgets. Also, VTK Qt widget plug-in works just in release mode compilation and this causes additional warnings during the tests. The project risks root in different factors such as software related, device-component originated, human errors, or environment. A partition of risks that were classified to be in association with software is shown in Figure 2.9.

In collaboration with medical experts, all identified risks were categorized in Failure Mode and Effects Analysis (FMEA) matrix (Table 2.2). Red blocks are unacceptable zones and the associated risks could cause sever harm on the patient or frequently happened. This type of risks requires tighter security measures during the software design. For example, the operator may enter a wrong X-ray generation parameter which leads to harmful over (or under) shoot exposure on the patient. To minimize this risk: 1) all exposure parameters on the user interface should have measurement unit, 2) the font and its size should be legible 3) the parameters' order on the screen should be static such that it reduces the chance of human mistakes, 4) there should be

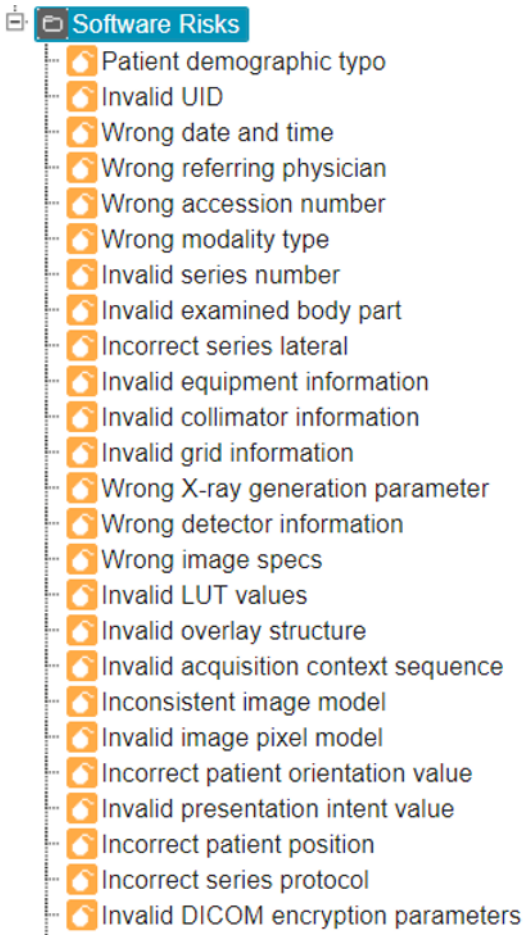


Figure 2.9 Snippet of risks in association with iBEX software. These risks could negatively affect on the device functionality and iBEX progress.

hard-coded boundaries on exposure parameters, 5) ask the operator to verify the exposure before starting examinations. Risks within the orange blocks are tolerable. For example, invalid Look Up Table (LUT) values may change the image appearance on the screen; however, most DICOM viewers have an option to manipulate LUT boundaries and physicians can adjust them on their device before deciding. Additionally, a verified DICOM encoder, like DCMTK, takes care about this issue. Finally, the green area shows the zones within which the risks happen remotely, or their severity is not critical. Example of this kind of risks are invalid grid information. This type of information is used to augment the DICOM file with device specification and does not carry therapeutic or diagnostic value.

The following strategies were contemplated to reduce harm:

Table 2.2
Risks FMEA matrix before applying any risk reduction mechanisms.

Likelihood	1 negligible	2 marginal	3 critical	4 very critical	5 catastrophic
6 frequently	1	1			
5 probable	2	22		23	
4 occasional			30	10	13
3 remote	19	16		5	4
2 improbable	13				
1 unbelievable	2				

Severity

- Apply standard compatible tools such as DCMTK.
- Require the user to verify steps.
- Hard code DICOM-predefined constants within the source code.
- Design user interface to reduce the chance of miss interpretation.
- Provide detailed documentations regarding device technical specification.

Based on the expert bodies' judgment, the above approaches could reduce the red zone risks' severity or likelihood and put them within the tolerable boundaries. The FMEA matrix of iBEX after risk management activity is shown in Table 2.3. One can see there is no risks within intolerable region.

The functionality and reliability of iBEX were evaluated on a novel TDI full body scanner. For each component, a device plug-in was developed and a denoising filter implemented to reduce the image noise. Sample output of a step phantom is shown in Figure 2.10.

A summary of the runtime resource consumption is given in Table 2.4. After successful login and before activating any process, a snapshot of allocated resources is captured. This snapshot constitutes the base line and during the execution, other snapshots were captured to compare the resource consumption over the time. A sudden

Table 2.3
Risks FMEA matrix after applying risk control mechanisms.

Likelihood					
6 frequently					
5 probable					
4 occasional					
3 remote	3	5	23		
2 improbable	18	52	8	13	
1 unbelievable	16		14		1
	1 negligible	2 marginal	3 critical	4 very critical	5 catastrophic
	Severity				

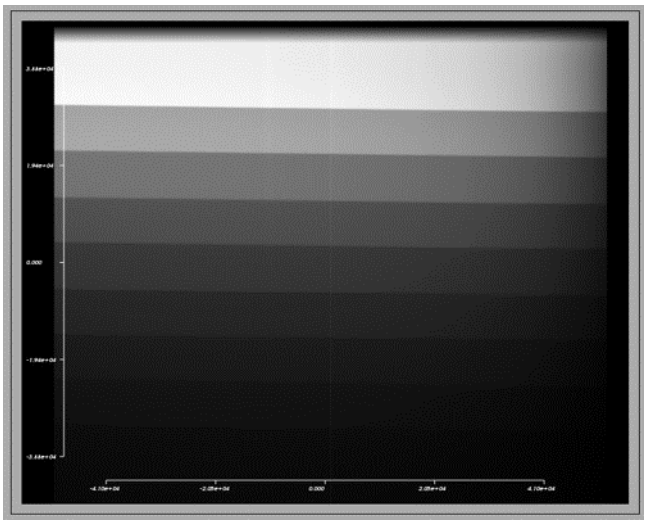


Figure 2.10 Sample output of a step phantom. In this study, an eight-level step phantom is used to mimic human body at different thicknesses.

memory and CPU peak was observed in quick-scan snapshot. This is because the image processing and device communication happened simultaneously, and they are resource hungry processes. Comparing CPU and GPU utilizations clearly depicts that most of the processing tasks are done by CPU rather than GPU. This is because the fact that VTK and CTK widgets were compiled for CPU.

A code coverage tool was applied to measure the percentage of the iBEX source code that involved in action on the prototype scanner. For this, five different execution scenarios were contemplated and for each scenario the percent of involved function to the total number of functions was measured (Figure 2.11).

Table 2.4
Memory, CPU and GPU utilizations reflect the runtime resource management quality criteria.

Snap	Memory (MB)	Peak CPU(%)	GPU utilization(%)
Baseline	22	17	2.8
PACS transaction	25.42	14	4.8
RIS transaction	25.43	11	3.2
Patient registration	26.34	4	3.7
Quick scan	231.88	25	5.2

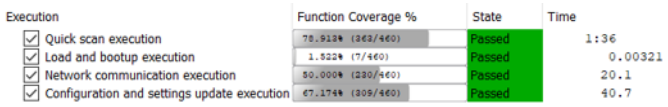


Figure 2.11 iBEX function coverage on prototype device. Number of iBEX functions involved in execution depends on the execution path.

The load and boot process required the minimum amount of functions to complete. This process was finished in almost 300 milliseconds. Network communication (i.e. PACS/RIS communication workflow) required almost 20 seconds and configuring the device settings took 40 seconds. The maximum amount of iBEX which is approximately equal to 80% of iBEX got involved to accomplish a quick scan. From another perspective, Figure 2.12 summarizes the condition pathways of each execution scenario

2.4 Discussion

iBEX is an open source and platform-independent software for digital X-ray scanners. In contrast to [59], iBEX is developed to be platform-independent-realization of an X-ray workstation software package. Its development process was adopted to be compatible with IEC 62304, ISO/IEC/IEEE 29119-5 and ISO 14971 medical device

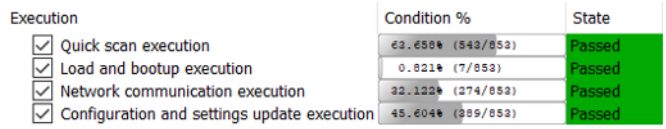


Figure 2.12 Percent of if-else statements passed to accomplish each execution scenario depends on its complexity.

standards. Project documents, samples and help files were prepared in RUP templates. These documents, when accompanied with test results of the whole X-ray scanner, could provide evidence of compliance with certifications like FDA or CE. Nonetheless, we must emphasize that iBEX is not evaluated nor applied for pre/clinical and human examination. Our current iBEX implementation is limited to phantom imaging and development of advanced image processing techniques required by our line scanner setup. Manufacturers and researchers can continue customizing and developing both iBEX and their device in compliance with pertinent regulations to finish up with a medical grade certified X-ray device. Open source community should take notice the importance of the full iBEX package, including the complete set of development notes, developer guide, manuals, risks analysis, source code test results, and examples. Similarly, any contribution must come with necessarily additional user and developer documentations. In the future, changes may happen to either iBEX core itself or new plug-ins will be developed. Whenever iBEX core is updated, source code tests and risks analysis should be re-evaluated and reflected on the relevant project documents. In this way, iBEX core consistency is preserved, and the updated documents can be used to provide evidence of compliance with medical standards. In the same manner, suspicious plug-ins will be detected by inconsistent risk analysis or incomplete source-code-test results. Any discrepancy between documented claims and compiled code could be a sign of a vulnerable contribution and the other open-source community members could report it to fix the issue or completely remove it from the project repository. The idea of plug-in extension is not novel to the medical imaging. For example, in Magnetic Resonance Imaging (MRI) modality, manufacturers with an agreement provide an extension mechanism for researchers to integrate their custom sequences and/or reconstruction algorithms. Usually, in those devices, the researchers' contributions execute in a controlled manner and they are utilized only after the institutional approval (which may include the ethical review by a formal body for animal and/or human research) has been acquired. Likewise, iBEX core utilized the Dependency Inversion Principle (DIP) [43] in order to isolate the plug-ins from the functionalities that are critical for the safety of the device. Communication in between plug-ins and core is formalized with IDevice and IAlgorithm interfaces and the core has the data flow control in hand. Additionally, each IDevice plug-ins runs within their

dedicated isolation environment and they cannot interfere with each other's operation. This separation protects the whole system against outlier device plug-ins and ease the debugging process. The IAlgorithm plug-ins are also sandboxed; furthermore, they receive a separate execution thread. Thus, the core is protected against system blocking image processing algorithms. Despite all these arrangements, there is still a chance for abnormal incidents. It is the responsibility of the developer to accomplish comprehensive testing, which includes plug-in tests, plug-in and core junction tests, and whole scanner device tests. Finally, for protecting the system against unauthorized plug-ins, a plug-in certification and signing mechanism could be added to plug-in manager. This protection mechanism would be complex and from our point of view it is in contrast with 'open-to-extension' spirit of iBEX; however, for commercial devices it seems to be unavoidable. Among the list of exerted packages, VTK, SQLite, CTK and Log4Qt are open source and freely available on their corresponding owner's websites; but, Qt and DCMTK are exception in this list. Qt is distributed under both open source freeware for noncommercial projects and commercial licenses. iBEX developers, specifically commercial users, should adhere to legal obligations enforced by the licensing agreements. Currently, DCMTK free evaluation period is for four months and any further use requires a full license agreement. Also, some device component manufacturers may ask for a non-disclosure agreement (NDA) to protect their Application Programming Interfaces (API's). In such situations, developers should avoid to publicly share their plug-ins. For example, in our full body scanner case, Gulmay power generator API was under protection and its plug-in's implementation file (i.e. .cpp file) was removed before uploading into the repository. iBEX plug-in mechanism breaks the limitation of as-is X-ray machine and turns it into a plug-integrate-play (PiP) device. In other words, researchers, based on their budget and workload, can select their own scanner's configuration and optimize its design. The details of how to develop a new plug-in and its source code is available on GitHub [42] repository readme section. The repository contains both device and algorithm plug-ins which could be used as template for developing plug-ins. iBEX is an ongoing project and open to improvements. As seen in Figure 2.11, execution time of configuration and settings update is not negligible and could be improved. There are some limitations in the current release. For example, there is no platform independent programming library available for writing a CD or

DVD. So, this tool is not yet implemented in iBEX. Medical printers' specifications differ from conventional printers and to best of our knowledge there is no platform independent medical printer emulator or library. However, in the future these tools are feasible to be developed by the community.

3. A WORKFLOW FOR ENSURING DICIOM COMPATIBILITY DURING RADIOGRAPHY DEVICE SOFTWARE DEVELOPMENT

In modern therapeutic and diagnostic devices, incompatibility with Digital Imaging and Communications in Medicine (DICOM) standard is a serious risk. DICOM incompatible radiology devices could cause undetected image lost, impose extra time and cost on health centers, and even wrong patient treatment. On the other side, there is a rich literature of medical standards that clarify the best practices for producing safe and effective medical software. However, these standards do not specifically provide tools to deal with all the relevant DICOM compatibility issues in a specific case. The objective of this study is to introduce a systematic software development workflow that complies with medical standards and ensures DICOM compatibility of the new or upgraded radiology software projects.An improved workflow for radiology device software development is introduced. In this approach, DICOM compliance gets the highest priority and whole software project is organized around it. Software requirement analysis, risk evaluation, and test management are organized systematically to make the final device DICOM compatible. This conceptual framework was developed during the R&D work towards a novel radiography device and it could be employed as a roadmap in other medical imaging hardware/software projects. The quantitative assessment of the open source project showed that the developed methodology mitigated the DICOM compatibility risk of the developed software, and its systematic evaluation complied with all the required medical standards. DICOM standard is vast and open for interpretation. Mistakes during design and implementation of a DICOM compatible device software could end up with financially expensive flaws and risks that might seriously harm the patients. The newly introduced systematic workflow proposed in this chapter considers the DICOM compatibility as the core issue of a radiography software project and controls it systematically during the project duration.

3.1 Introduction

Risk is intrinsic to medical devices and may negatively affect their expected operations. For this reason, it is a point of concern of many medical standards, such as IEC 82304 (general requirement for product safety) [60], ISO 14971 (risk management) [40], and IEC 60601/61010 (safety and performance) [61], which all encourage manufacturers to establish processes that commensurate with the risks of developing their medical devices and prove their safety and effectiveness. Mathematically, multiplication of the probability of occurrence of a harm with its severity is equal to the risk associated with that harm [62]. For example, in radiology departments, there is a chance to assign acquired digital X-ray images to a different patient [63] or store it without a sound and complete set of credentials. In such circumstances, the examination outputs might be lost, and the patients may require further examinations. Repeated exams would impose extra burden on the healthcare centers and increase the harmful X-ray dose to the patients. These instances are not uncommon in daily clinical practice [64]. The risk associated with losing valuable data in radiology departments after finishing the examination has been reported to be relatively high [64,65]. Digital Imaging and Communication in Medicine (DICOM) standard [66] was introduced indeed to prevent and resolve these and similar risks. Since the introduction of DICOM about 30 years ago, it became the driving force for standardizing the workflows of medical imaging procedures. During this time, DICOM is evolved for all parts of digital image acquisition, transmission, printing, and processing. This mission expansion made DICOM more complex and open for interpretation [54, 64]. As a result, despite the recent advances in digital imaging, most healthcare centers keep losing their DICOM files, while some of these losses going completely unnoticed. One common reason for this problem is the scanner device incompatibility. Despite the vendors DICOM conformance claims, they still provide their own implementations (and often interpretations) of DICOM in their image files, which might cause errors in patient reports at the end [64]. DICOM standard provides a collection of what data should be acquired and what should be done with it; while, it excludes methods and techniques for how to get to such data. On the other hand, standards like IEC 62304 (Medical Device Software – Life Cycle Processes) [37] and ISO/IEC/IEEE 29119 (software testing) family exist, which provide a

pathway for how to make a safe and effective medical device software that could either be an independent software or part of a device. Implementation of these standards was the subject of several studies. As an example, formal methods and Abstract State Machine (ASM) were proposed to develop a medical device and then were applied for developing hemodialysis and optometric measurement devices' software [67, 68]. Even though these approaches have a good performance in identifying and controlling the risk of a medical device, they still require a priori model of the medical system. In real world, this kind of models are available or achievable only for systems with limited complexity, and only a few variables and parameters [69]. However, in the case of complex systems, such as medical radiography scanners, and for complex issues like full DICOM compliance, more elaborate methods are needed. In this work, a new software development methodology is introduced as a systematic workflow for developing a radiography scanner software, to fill the gap between DICOM standard and other medical software development standards. This method initiates and organizes the software development tasks in accordance with all relevant medical standards and DICOM necessities. The method was developed during the development of an open source digital radiography workstation software project, named iBEX [70]. This software package is an open source radiology workstation for digital X-ray scanners, which was designed to be a generic workstation software suitable for development of various digital radiography components (e.g. X-ray detector, power generator, etc.) of different vendors and integrate them into its core module during boot up. The focus of this work is to provide a general framework for DICOM compatible radiography devices. The proposed methodology could be used to customize and extend the iBEX platform itself or develop a new radiography workstation software. In the following sections, the relevant background is given in further detail at Section 3.2, the new methodological approach is described at Section 3.3, the implementation of this proposed method in iBEX is presented as a case example at Section 3.4, and finally, the discussion is presented at Section 3.5.

3.2 Background

Nontrivial software projects are managed within the context of a Software Development Life Cycle (SDLC) plan, which consists of a set of steps. An example lifecycle is shown in Figure 3.1. First, an initial plan is documented, then a business model of this plan is created. Following that, in a comprehensive effort, requirements of the business plan are identified, analyzed and then implemented. The achievements are tested and assessed for performance and requirement coverage. In practice, project manager may decide to omit or mix some stages into each other or rearrange the order of them. Satellite tasks, such as project management, environment management, and configuration management are done throughout the project [71] to control and monitor the whole process. For complex projects, this ring of operations is repeated multiple times. In each iteration, a fraction of project is prepared and incrementally added up together till the whole system is completed [72]. Medical standards do not limit the SDLC options for medical projects. They strongly suggest applying a systematic approach, like the one shown in Figure 3.1 (with some differences in terminologies). For example, in [37,72] a V-model SDLC was proposed. In this model, steps were arranged in system-level, component-level, and implementation-level. Each level was followed by validation-and-verification of that level, and whenever a level was finished, and before the next level got started, its validation-and-verification routine was conducted. Medical standards emphasize more on the following: 1) requirement classification, 2) software validation and verification, and 3) risk management [62,63,73]. In the remaining of this section, these issues are discussed in more detail.

3.2.1 Requirement Classification

The requirements analysis is normally accomplished in multiple levels of details and may be updated during the project progress [74]. The output of this step is a set of documents, which forms the final output's shape. IEC 62304 and IEC 82304 clarify three types of requirements: 1) design inputs are the interpretation of customer needs into formally documented medical device requirements, 2) software requirements are

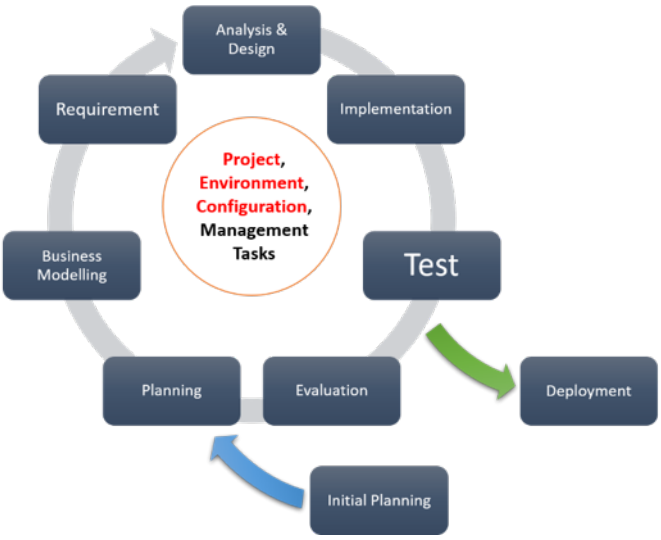


Figure 3.1 Software development lifecycle (SDLC) consists of a set of steps. Project manager may change the arrangement of these steps.

the formally documented specifications of what the software does to meet the customer needs and the design inputs, and 3) system requirements are the documentation of the functionality for intended use of the device and includes all hardware, software, and peripherals. These three types help to subjectively organize the requirements in appropriate documents and formats.

3.2.2 Software Validation and Verification

Software verification looks for consistency, completeness, and correctness of the software and its supporting documentation, as it is being developed [25]. Software testing is one of many verification activities intended to confirm that software development output meets its input requirements. Other verification activities include various static and dynamic analysis, code and document inspections, walkthroughs, and other techniques [75]. Software validation is "confirmation by examination and provision of objective evidence that software specifications conform to user needs and intended uses, and that the particular requirements implemented through software can be consistently fulfilled" [75]. In practice, software validation activities may occur both during, as well as at the end of the software development, to ensure that all requirements have been fulfilled. Since software is usually a part of a larger hardware system, the validation

of software typically includes evidence that all software requirements have been implemented correctly and completely and they are traceable to system requirements [37,76].

3.2.3 Risk Management

IEC 62304, IEC 14971, and the other guidelines like Food and Drug Administration (FDA) Current Good Manufacturing Practice (CGMP) [77] require manufacturers to make risk management as the core competency of their development life cycle to prove the safety-critical properties are hold, and to guarantee conformance to abstract specification of a safe device operation. In medical standards risks are classified according to severity of potential harms: Class A in which harms can cause no damage to health; Class B in which harms cause non-serious injury; and in Class C harms could end up in death or a serious injury.

Risk analysis and classification should be done hierarchically through all internal sub-systems and external packages. This effort is dynamic and depends on thoroughness - higher risks need more effort [62, 74]. The variety of medical devices reaches this goal in a variety of approaches and there is no "one size solution to fit all". The purpose of the ISO/IEC/IEEE 29119 series of software testing standards is to define an internationally agreed set of standards for software testing that can be used by any organization when managing or performing any form of software testing. Specifically, ISO/IEC/IEEE 29119-5 [38] explains the main concepts and application of Keyword-Driven Testing. Keyword-Driven Testing is a test case specification approach that is commonly used to support test automation and the development of test automation frameworks [78]. In principle, Keyword-Driven Testing can be applied at all testing levels (e.g. component testing, system testing, etc.) and for various types of testing (e.g. functional testing, reliability testing, etc.). The fundamental idea in Keyword-Driven Testing is to provide a set of *building blocks*, referred to as keywords, that can be used to create manual or automated test cases without requiring detailed knowledge of programming or test tool expertise. The keywords are comprehensive enough so that most, if not all, required test cases can be entirely composed of these keywords. The

vocabularies included in these dictionaries or keywords' libraries are, reflection of the level of abstraction, and human readable and not confined to any standard computer programming language. However, during automated testing, each keyword needs to be implemented in software language.

3.2.4 Project Planning

The systematic approach to a medical software project requires a thorough planning. Depends on the project scale, time, and resources; documents such as project plan, development configuration management plan, integration plan, verification plan, risk management plan, and quality management plan may be prepared and continuously updated by the project manager. Effective planning requires a good knowledge about the project in the first place. Specifically, for radiography projects the American College of Radiology (ACR), the American Association of Physicists in Medicine (AAPM), and the Society for Imaging Informatics in Medicine (SIIM) provide a set of technical guidelines for practitioners which can be used as a good source for getting introduced to radiography scanner workflows and systematic requirements. ACR–AAPM–SIIM practice guideline for digital radiography [34,41,79] defines flexible equipment, data manipulation and management, and quality control (QC) and quality improvement procedures. It includes guidelines for any digital image relevant medical-system, from a single-modality or single-use system to a complete PACS infrastructure. It defines goals, qualifications of personnel, equipment guidelines, specifications of data manipulation and management, quality control and quality improvement procedures that should result in high-quality radiological care. ACR–AAPM–SIIM parameter for electronic medical information privacy and security [80] recommends actions for the protection, privacy, security, and integrity of recorded patient information while allowing appropriate access for care and management of patients. Policy and procedure recommendations are provided, and the available tools to ensure privacy and security are described. The other potential source for planning is DICOM standard. This standard contains thousands of tags which are logically modeled into objects and relation in between them. The details related to this model is documented at part three of

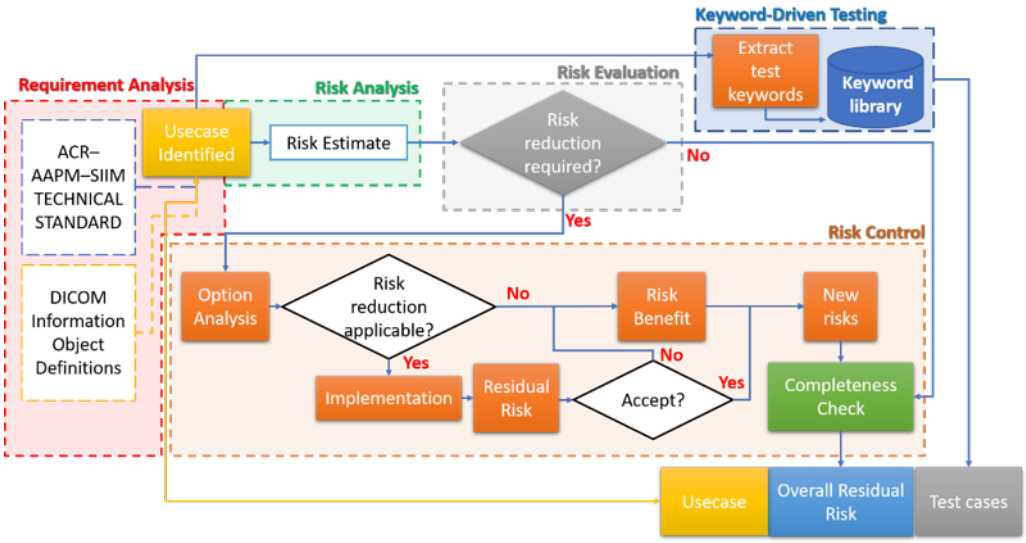


Figure 3.2 New workflow for DICOM compatible software design. Requirement analysis stems in DICOM and relevant standards.

DICOM standard - under the title "The Information Object Definition" [66]. This model provides list of essential and optional data within DICOM compatible modalities. Project manager could use DICOM model for getting an overview of all data flow within the project.

3.3 Material and Methods

The software unit of DICOM compatible radiography scanner, is responsible for orchestrating dozens of parameters, continuously monitoring the state of the device for detecting and reacting to faults, communicate with healthcare infrastructure (or other modalities and machines), store and retrieve data; all in DICOM compatible way and commensurate with all the relevant medical standards. A generic form of a new work plan for developing radiography scanner software is shown in Figure 3.2. The rationale of this new method is to bring requirement analysis, risk management and test management tasks under the same umbrella and proceed with them at the same pace. The other trait of this work plan is its inception point. This work plan starts directly from DICOM Information Object Definitions (IODs) [66] accompanied with ACR-AAPM-SIIM guidelines. This last point is important in two ways: first,

in compare to the other well-known software engineering methodologies [81], it brings the DICOM compliance requirement into the focus point, rather than considering it as a system requirement or functional specification. This emphasize on DICOM at the beginning reduces the chance of incompatibility risk at the end of the project. Second, one should consider that the DICOM standard is a thousands of pages long document and in some sections, it is hard to understand [54]; however, starting directly from the relevant section at the beginning of the project could speed up the design process. Each data and relation within these resources, is a potential usecase (requirement). For each identified usecase two tasks are started simultaneously:

1. 1. Extract Keywords for Designing Test: With respect to level of abstraction, the keywords are categorized into high-level (user-domain-layer) or low-level (test-interface-layer). In some cases, one or more middle layers are also added. Then, the classified keywords are used for designing automatic or manual test scenarios which are executed inside a test framework [38].

2. 2. Risk Management:The risk-class associated with the usecase is evaluated first and if it was needed, methods for risk reduction is contemplated. Generally, there are three strategies to manage risks: a) avoid risk by safe design, b) protective alternatives inside device, c) safety information and warning on the device. After applying the selected strategy (or combination of strategies), the residual risk is reevaluated. If it is not acceptable, then further information is gathered to determine if the medical benefits outweigh the residual risk. If this evidence does not support the conclusion that the medical benefits outweigh the residual risk, then the risk remains unacceptable and project is terminated (which is not a case for radiography scanner). The applied strategy for controlling a risk has the chance of arising new risks or affect the other risks. In that case, the same steps repeat for those risks one more time- till all the risks associated with each usecase are revisited.

The outputs of the workflow include risks, usecases, and tests charts. The format and the content of the documentation are determined by the project manager

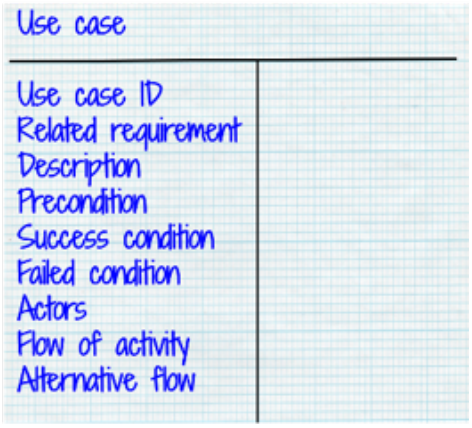


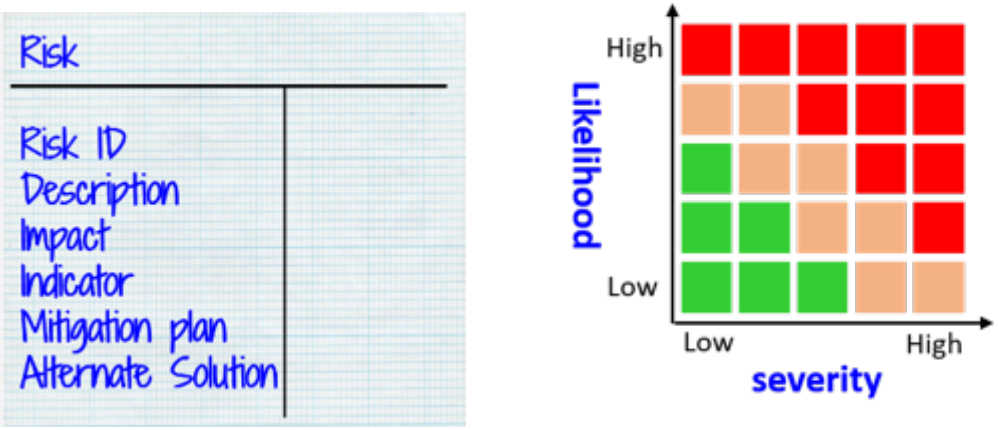
Figure 3.3 Usecase template. Recording the workflows beside usecase makes tracking easier.

and must be compatible with IEC 62304. A sample usecase chart is shown at Figure 3.3. In this chart, each usecase has a unique ID. Beside the usecase description, activation condition, successful situations, actors, and failure mode states are also recorded. Not all fields are required to be available on all usecases; but, recording this information eases the tracking of the requirements during the project. Details of provisioned risks are recorded in risk list document. A sample chart of a risk is shown in Figure 3.4 A. In this sample, plans to mitigate risk and matrices to measure the risk (if available) should be provided. At the end of the document a summary of the risks should be provided. In this summary, all risks before and after mitigation are summarized in Failure Mode and Effects Analysis (FMEA) matrix [82] (Figure 3.4 B). These matrices facilitate the project members to monitor the efficiency of the risk management strategies.

The test document should be compatible with ISO/IEC/IEEE 29119 – part 5. In this way, each identified usecase keyword, is a potential test case, but still it may require to be grinded in multiple levels (Figure 3.5). The domain layer is the most abstract layer and interface layer is the most detailed one. In between these two, one or more intermediate layers also could be added.

Finally, tests and their counterpart usecases should be recorded in traceability matrices [83].

The execution of the proposed workflow depends on the project methodology.



(a) Risk chart template (b) FMEA

Figure 3.4 Risk template includes details associated with each risk and Failure Mode and Effects Analysis (FMEA) provids a summery of all risks.

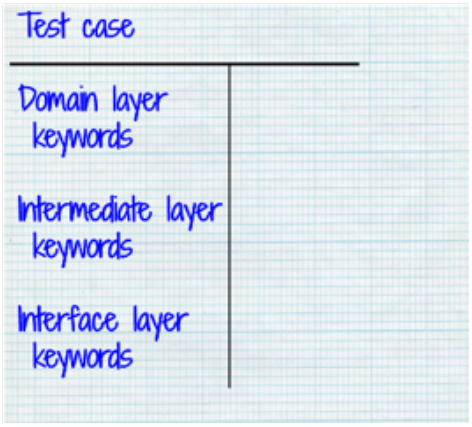


Figure 3.5 Keyword tests template. Each test may be designed in multiple layers of abstraction.

We suggest applying the proposed workflow in the context of an iterative and incremental methodology. For example, a modified Rational Unified Process (RUP) [84] is shown in in Figure 3.6. In this diagram, the size of each box represents the amount of effort spent on each activity at that time slice (i.e. inception, elaboration, construction and transition). The requirement analysis, risk evaluation and test management are inseparable and carry the same level of importance through the whole project (except the last step in which the project is almost finished). Within each time slice, there are one or more iterations. Number of these iterations are determined by environmental factors and managed by project manager.

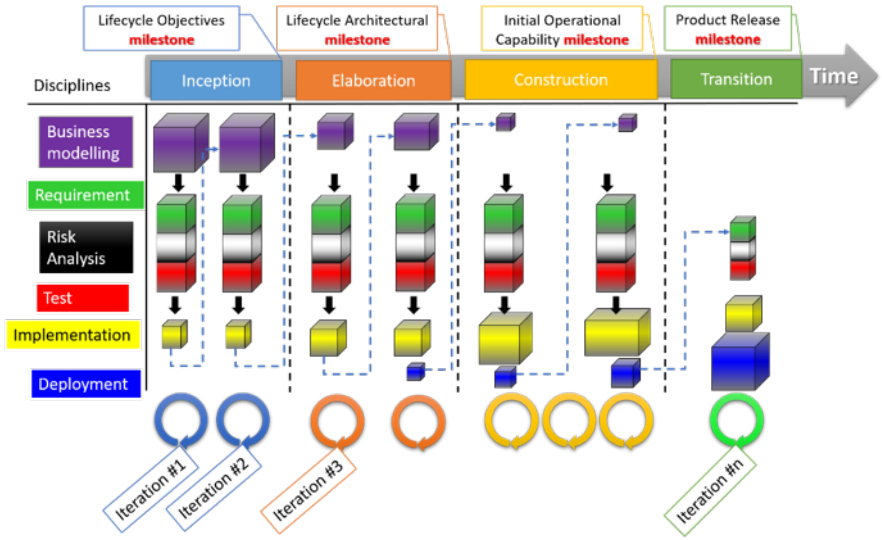


Figure 3.6 A modified RUP process is a candidate methodology for implementing the proposed workflow. In this workflow the requirement, the risk, and the test analysis are done simultaneously and at the same level all the time.

3.4 Results

The new workflow was developed in the course of iBEX open source radiography scanner workstation project. DICOM requirements are listed in Table 3.1. Some of the modules in this table are mandatory to be present (shown by 'M'); while, some others are user optional, which are marked by 'U'. The device specifications, workflows, and control mechanisms as described in ACR–AAPM–SIIM guidelines.

Each item of the Table 3.1 and Table A.1 represents a requirement and should be hammered during the project development. For example, in Table 1 the 'Patient IOD' is a 'keyword' for a usecase in which the patient demographic information 'must' (the M letter enforces its existence) be collected. This word is automatically added to the keyword library and during the risk analysis and evaluation, all risks that are associated with patient information are identified and categorized. As shown in Figure 3.7, 'Patient demographics typo' is an example of such a risk. These risks are analyzed for severity and then approaches to eliminate them or at least reduce the harm are devised. The risk analysis document of iBEX gives a summary of the risks before applying any risk-control mechanism (Table 2.2). There are considerable number of threats within the red zone. These are class C risks and require deeper consideration. For example,

Table 3.1
Digital radiography (DX) IOD of DICOM. Entities marked with 'M' are mandatory and entities marked by 'U' are user optional.

Information Entity (IE)	Module	Usage
Patient	Patient	M
	Clinical Trial Subject	U
Study	General Study	M
	Patient Study	U
	Clinical Trial Study	U
Series	General Series	M
	Clinical Trial Series	U
	DX Series	M
Frame of Reference	Frame of Reference	U
Equipment	General Equipment	M
Image	General Image	M
	Image Pixel	M
	Contrast/Bolus	U
	Display Shutter	U
	Device	U
	Intervention	U
	Specimen	U
	DX Anatomy Imaged	M
	DX Image	M
	DX Detector	M
	X-Ray Collimator	U
	DX Positioning	U
	X-Ray Tomo Acquisition	U
	X-Ray Acquisition Dose	U
	X-Ray Generation	U
	X-Ray Filtration	U
	X-Ray Grid	U

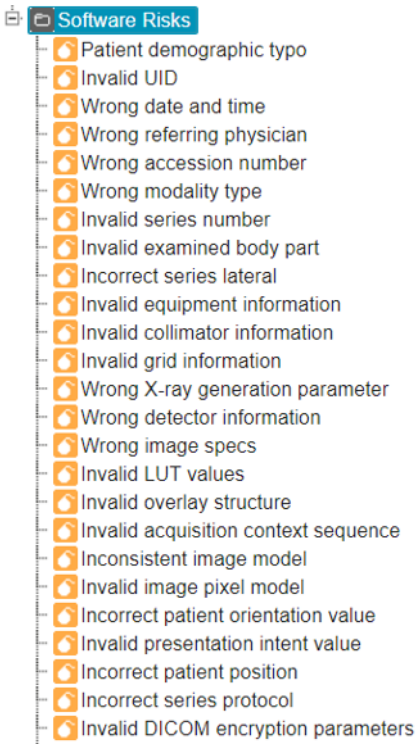


Figure 3.7 Software risks associated with identified usecases. Tracking the risks and their connections require a good development skill and tool to mange the interconnections.

operator may enter a wrong X-ray power parameter, which might lead to over or under exposure of the patient. To minimize this risk, all exposure parameters on the user interface should have a clear indication of the measurement unit, the font size should be legible, and the parameters' organization on screen should be such that it reduces any chance of mistake. Finally, the operator should be asked to verify the exposure parameters before starting it.

iBEX packages diagram is shown at Figure 3.8. In this diagram the core package is responsible for fundamental requirements (e.g. registration workflow) and in collaboration with plug-in engine and through IDevice and IAlgorithm interfaces the core functionalities could be extended. IDevice was designed for standardizing the communication between the core package and peripheral devices. IAlgorithm was an interface in between the core and external image processing and filtering plugins. The proposed method suggests progressing through requirement analysis, test design, and risk assessment at the same pace. By this approach, it would be possible to keep all the traceability matrices updated throughout the project. In Figure 3.9 a trace between

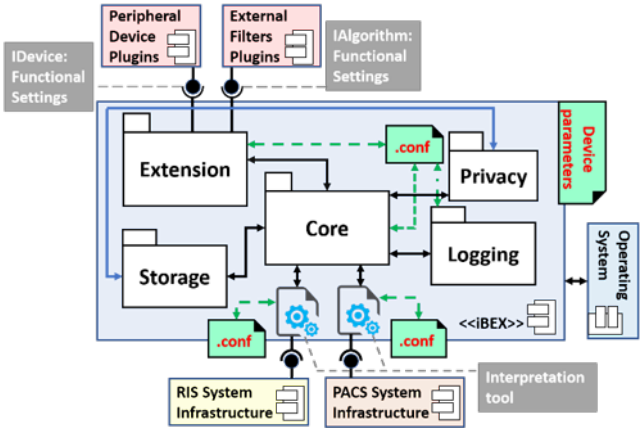


Figure 3.8 iBEX architecture package diagram. Packages are designed to fulfil the system requirements.

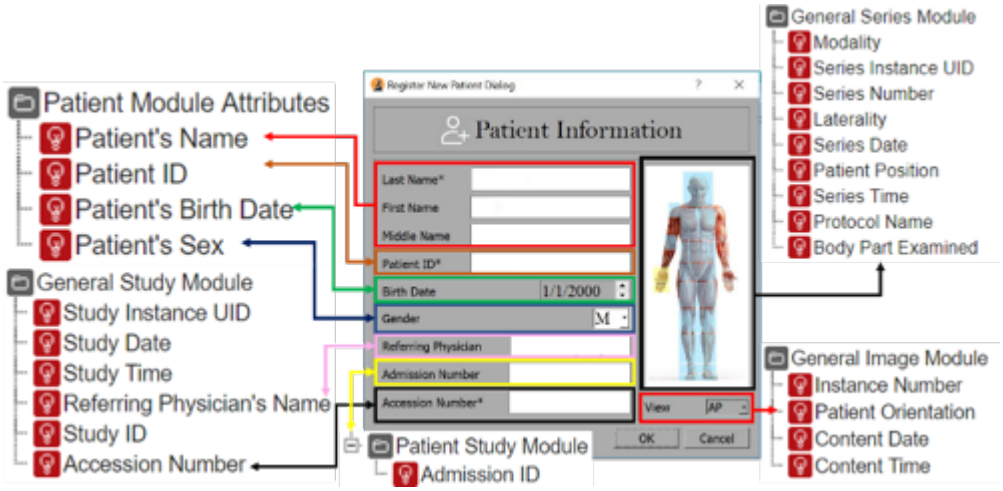


Figure 3.9 Each part of the final output of iBEX is traceable to the requirements definition. This traceability from design to final output represents a high quality product.

the output user interface and DICOM modules (i.e. usecases in requirement analysis document) is shown. Finally, the test keywords were extracted from contents of Table 3.1 and ACR–AAPM–SIIM workflows. Keywords were organized in two layers: the domain layer and test interface layer. Each noun in IOD list was a candidate test term: 'Patient name' test, 'Image Row' test, 'Generator KVP' test, and so on. Verbal phrases such as 'patient registration', 'device calibration' and 'PACS communication' were potential usecases with a one or more domain layers tests.

3.5 Discussion

Medical software development is tightly regulated, and successful projects are required to show evidences that prove the necessary standards have been applied during the project development. Specifically, medical imaging systems are required to comply with DICOM standard. In this work, a new software development approach was proposed, which aims to bridge the gap between medical manufacturing standards and DICOM standard. iBEX is an open source digital X-ray scanner workstation software [70] which was designed based on the proposed method. Its designed to be extensible and vendor neutral. These design-inputs, accompanied with DICOM model and ACR-AAPM-SIIM guidelines, were formalized within the software architecture document (i.e. 4+1 view architecture template [85]). This document includes major requirements of the system from logical, process, development and physical points of view. One of the major sources of the DICOM compliance problem is differences in the infrastructure implementation of the health centers. Each health center has strategies and requirements to select among the DICOM tags and communication options. These options provide a vast range of possibilities that is hard to handle. To mitigate the situation, iBEX uses interpretation tools. These tools based on a configuration file (.conf for iBEX) act as a translator in between PACS/RIS infrastructure and core package. They fill up the missing DICOM tags with default values or change the values to make it suitable for the Core package. Designing effective test scenarios require years of experience. However, the noun-and-verbal phrases technique mentioned in the section four could keep the process on track. For iBEX, in addition to keyword driven testing, which are designed to assess the code functionality, the source code quality was also assessed regarding the memory consumption, execution time, and code coverage yardsticks. The complete test list and their results are available at [70].

4. CONCLUSIONS AND FUTURE WORKS

Modern radiology scanners are versatile and complex devices. The workstation software is responsible for orchestrating this machine's all components, expected to be RIS compatible, provide tools for viewing and processing studies, manage local storage, and synchronize with PACS server. Finally, all outputs and communications should be DICOM compatible. iBEX is the first open source medical radiology workstation software tool with new promises for researchers and developers. It provides two ways of extension; one for integrating devices and another for integrating image processing algorithms. Using off-the-shelf components (e.g. power source, detector, etc.) from vendors (even local ones), then developing iBEX plug-ins for those components and integrating them into iBEX, one can have a device which exactly meets the requirements. This manufacturing paradigm which we called Plug-integrate-Play (PiP) has the following advantages:

- Democratize the device manufacturing. The current manufacturing is closed model and devices are a compact devices. Healthcare institutes have limited options for asking for customization.
- Reduce the device costs. iBEX is a freeware and publicly available. The only cost is developing some plug-ins which is relatively much less than developing a complete system from scratch.
- Potentially more immune against bugs and inefficiencies. iBEX introduces the medical software into open source community. In non medical applications the community has proven its potentials and brought successful results.
- Ease the researchers effort: in contrast to commercial products, iBEX is open for external filtering plug-ins. This is trait is valuable for developing new image processing filters.

iBEX is developed based on C++/Qt and does not depends on any specific

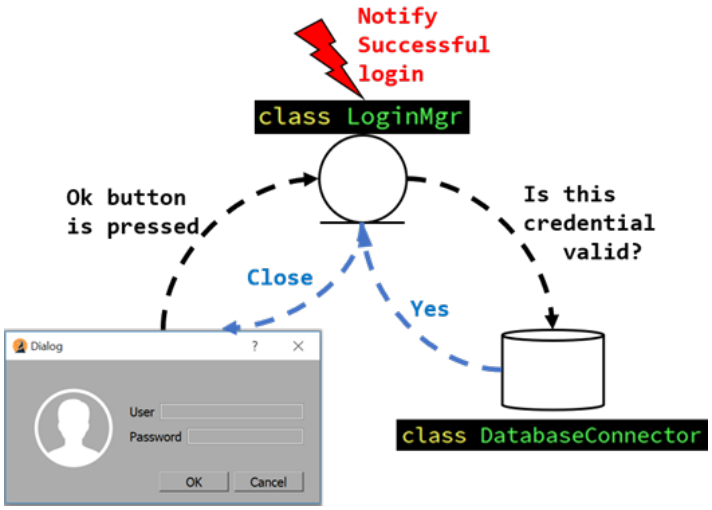


Figure 4.1 MVC pattern consists of model object, view object and control object. Transactions in between these objects are shown for user signing-in operation.

environmental parameters. This feature makes it easy to port the source code into any execution environment and architecture. All units of iBEX are developed based on Model-View-Controller (MVC) design pattern [86]. In this pattern, the view is responsible only for user interactions. When user interacts with the view objects, it sends a message to the controller class, the controller then checks the model (e.g. database) and based on the state, decides and updates the view and/or informs the other controllers. Example of this cycle for user sign work flow is shown in:

When the user enters his/her credentials into the sign in dialog, the information is sent to the control class (i.e. LoginMgr). Then, this control object asks model for credentials validation (i.e. DatabaseConnector). If the model can find the data, then the LoginMgr close the view dialog and informs all possible listening controllers with the event of a new successful login. This paradigm governs all units of the iBEX core and almost all use cases follow the same paradigm (Figure 4.1).

over the course of iBEX development, a new methodology is introduced. This methodology is designed for projects that are needed to be DICOM compatible. The proposed methodological framework brings DICOM compliance into the focus during the design and implementation of a software project. The notable achievement of this framework is a great reduction in DICOM incompatibility risk of the final product.

Additionally, this framework would help software designers to start their project from the right point, otherwise, they may need to spend a lot of time and effort to find an inception point for entering into DICOM (and the other relevant medical standards). iBEX is the first software project that employed this methodology in-house; and both software and methodology are presented to the medical imaging research community for collective development in the future, within the spirit of democratization of medical imaging hardware/software development [42].

There are some concerns that should be considered :

1. iBEX follows software engineering best practices and methodologies. All the documents that are generated during the development are publicly available. However, it is the responsibility of the users to re-evaluate and accomplish the tests after making any change or update.

2. iBEX is distributed as-is. It is on the final user to apply for certifications (if needed). However, iBEX contributions come with full access to source code, documents, and test results. These satellite documents could be used in applications for a medical certification.

3. Any contribution to iBEX should come with full access to the design, development, and test documents. In this way the whole community would access to the details of the development and easily identify any suspicious contribution.

Finally, iBEX is on its first steps and there are things to do. In the future, integration with medical printers, tests on different RIS, HIS and PACS systems could be useful. In the current version, iBEX extension mechanisms accepts plug-ins without any control at runtime. However, implementing a signed plug-ins mechanisms could protect the system against suspicious malware. Also, collecting all testified plug-ins in one pool could ease the extension for new designs.

REFERENCES

1. Seeram, E., *X-ray Imaging Equipment, 1st ed.*, Ill., U.S.A.: Springfield, 1985.

2. Wolbarst, A., *Physics of Radiology, 1st ed.*, Ill., U.S.A.: Madison Wis.: Medical Physics Pub., 2000.

3. Website, "Dap meter," online resources, Trust International Trading Co, 2020. Available: https://xraycomponents.com/components/dap-meter , last access: Feb,2020.

4. Sprawls, P., *Physical Principles Of Medical Imaging, 1st ed.*, Rockville, U.S.A.: Md.: Aspen Publishers, 1987.

5. Seibert, J., "Flat-panel detectors: how much better are they?," *Pediatric Radiology*, Vol. 36, pp. 173–181, Sep 2006.

6. Systems, L., "Lodox x-ray technology: An explanation," white apaper, Los Alamos, 143 Burton Street, Painesville, Ohio 44077, 2018. [Online]. Available: http://lodox.com/wp-content/uploads/2018/08/Exero-tech-specs.pdf.

7. "American film-starlet, judith allen, has a radiograph taken of her back in 1930," 1930. Available: https://twitter.com/WakeRad/status/877891223951994880.

8. Papp, J., *Quality Management In The Imaging Sciences, 5th ed.*, St. Louis: Missouri: Elsevier, 2015.

9. Website, "Zz medical," online resources, Trust International Trading Co, 2020. Available: https://www.zzmedical.com/analog-x-ray-supplies/x-ray-grids/x-ray-grids-for-digital-radiography.html , last access: Feb,2020.

10. Yaffe, M., and J. Rowlands, "X-ray detectors for digital radiography," *Physics in Medicine and Biology*, Vol. 42, pp. 1–39, Jan 1997.

11. MARTIN, M., "macmartin," online resources, macmartin, 2020. Available: https://mackmartin958519424.wordpress.com/2019/05/03/global-x-ray-flat-panel-detectors-market-2019-varex-imaging-toshiba-canon-trixell-analogic/ , last access: Feb,2020.

12. control eng., "macmartin," online resources, control eng., 2011. Available: https://www.controleng.com/articles/scanning-x-ray-detectors-for-dental-medical-imaging/ , last access: Feb,2020.

13. Fu, K., "D trustworthy medical device software," in *Institute of Medicine (US) Committee on the Public Health Effectiveness of the FDA 510(k) Clearance Process* (T, W., ed.), Washington (DC): National Academies Press (US): National Academies Press, Des 2011. Available from: https://www.ncbi.nlm.nih.gov/books/NBK209656/.

14. Deyle, S., A. Wagner, L. Benneker, V. Jeger, S. Eggli, H. Bonel, H. Zimmermann, and A. Exadaktylos, "Could full-body digital x-ray (lodox-statscan) screening in trauma challenge conventional radiography?," *The Journal of Trauma: Injury, Infection, and Critical Care*, Vol. 66, p. 418 – 422, Feb 2009.

15. Perks, T., R. Dendere, B. Irving, T. Hartley, P. Scholtz, A. Lawson, C. Trauernicht, S. Steiner, and T. Douglas, "Filtration to reduce paediatric dose for a linear slot-scanning digital x-ray machine," *Radiation Protection Dosimetry*, Vol. 167, pp. 552–561, Nov 2014.

16. Potgieter, J., M. de Villiers, M. Scheelke, and G. de Jager, "An explanation for the extremely low, but variable, radiation dosages measured in a linear slit scanning radiography system," in *Medical Imaging 2005: Physics of Medical Imaging*, Vol. 5745, (San Diego, California–United States), Proc. SPIE 5745, 20 Apr 2005. https://doi.org/10.1117/12.595216.

17. Whiley, S. P., H. Alves, and S. Grace, "Full-body x-ray imaging to facilitate triage: A potential aid in high-volume emergency departments," *Emergency Medicine International*, Vol. 2013, p. 6, Aug 2013.

18. Krans, B., "Lumbosacral Spine X-Ray: Purpose, Procedure, and Risks," 2017. Available: https://www.healthline.com/health/lumbosacral-spine-x-ray.

19. Beningfield, S., H. Potgieter, A. Nicol, S. As, G. Bowie, E. Hering, and E. Lätti, "Report on a new type of trauma full-body digital x-ray machine," *Emergency Radiology*, Vol. 10, pp. 23–29, Feb 2003.

20. Mulligan, M. E., "Imaging techniques used in the diagnosis, staging, and follow-up of patients with myeloma," *Acta Radiologica*, Vol. 46, pp. 716–724, May 2005. DOI https://doi.org/10.1080/02841850500215360.

21. Group, I. S. W., *Software as a medical device (SaMD): key definitions*, ch. Key Definitions, pp. 1–9. International Medical Device Regulators, IMDRF/SaMD WG/N10FINAL, 2013.

22. Schneider, P., and M. L. A. Hines, "Classification of medical software," in *Proceedings of the 1990 Symposium On Applied Computing*, Vol. 5745, (Fayetteville, AR, USA), pp. 20–27, 1990. doi: 10.1109/SOAC.1990.82134.

23. Bassen, H., J. Silberberg, F. Houston, W. Knight, C. Christman, and M. Greberman, "Computerized medical devices: Trends, problems, and safety," in *IEEE Aerospace and Electronic Systems Magazine*, Vol. 1 of *9*, (Fayetteville, AR, USA), pp. 20–24, 1986. doi: 10.1109/MAES.1986.5005202.

24. Sear, A. M., "Concepts and Issues in Health Care Computing," *The Journal of Ambulatory Care Management*, Vol. 10, no. 2, 1987.

25. Rakitin, and S. R., "Coping with defective software in medical devices," in *Computer*, Vol. 39 of *4*, (Washington, DC, USA), pp. 20–24, IEEE Computer Society Press, April 2006. doi: https://doi.org/10.1109/MC.2006.123.

26. et al., L., "High-confidence medical device software and systems," in *Computer*, Vol. 39 of *4*, (Washington, DC, USA), pp. 33–38, April 2006. doi: 10.1109/MC.2006.127.

27. Chen, Qiushui, Wu, Jing, Ou, Xiangyu, Huang, Bolong, Almutlaq, Jawaher, Zhumekenov, A. A, Guan, Xinwei, Han, Sanyang, Liang, Liangliang, Yi, Zhigao, *et al.*, "All-inorganic perovskite nanocrystal scintillators," *Nature*, Vol. 561, pp. 88–93, Feb 2018.

28. Korner, Markus, Weber, C. H, Wirth, Stefan, Pfeifer, Klaus-Jurgen, Reiser, M. F, Treitl, and Marcus, "Advances in digital radiography: physical principles and system overview," *Radiographics*, Vol. 27, pp. 675–686, May 2007.

29. Lee, Eunsin, Meyer, Juergen, Sandison, and George, "Collimator design for spatially-fractionated proton beams for radiobiology research," *Physics in Medicine & Biology*, Vol. 61, p. 5378, Jun 2016.

30. AndyC, "A free website for radiographers, sonographers and students," online, wikiRadiography, USA, Feb 2020. Available: http://www.wikiradiography.net/page/Radiographic+Protocol+Guide.

31. Alava, P., and J. Antonio, "Computer vision and medical image processing: a brief survey of application areas," in *Argentine Symposium On Artificial Intelligence (ASAI 2015)*, (Argentine), pp. 152–159, Sociedad Argentina de Informatica e Investigacion Operativa (SADIO), SEDICI, 1-4 Sept 2015.

32. A.Seibert, J., "Fundamentals of cr, dr and pacs," online, IAEA, California -United States, Feb 2012. Available: https://inis.iaea.org.

33. Samei, E., and M. J. Flynn, "An experimental comparison of detector performance for direct and indirect digital radiography systems," *Medical Physics*, Vol. 30, pp. 608–622, Mar 2003. doi: https://doi.org/10.1118/1.1561285.

34. ACR–AAPM–SIIM, "Acr–aapm–siim technical standard for electronic practice of medical imaging," online, Iowa Medical Society and Iowa Society of Anesthesiologists v. Iowa Board of Nursing, Iowa, Feb 2017. Available: https://www.acr.org/-/media/ACR/Files/Practice-Parameters/Elec-Practice-MedImag.pdf.

35. CDER, "Part 11, electronic records; electronic signatures - scope and application," online, FDA, New Hampshire, Aug 2003. Available: https://www.acr.org/-/media/ACR/Files/Practice-Parameters/Elec-Practice-MedImag.pdf.

36. 93/42/EEC, C. D., "Directive 2007/47/ec of the european parliament and of the council of 5 september 2007," online, FDA, EU, Jun 2007. Available: https://eur-lex.europa.eu/.

37. ISO, "Iec 62304:2006 - medical device software – software life cycle processes.," standard, ISO, Geneva, Jun 2006. doi: https://doi.org/11.040.01.

38. ISO/IEC/IEEE, "Iso/iec/ieee 29119-5:2016,iso/iec/ieee 29119-5:2016," standard, ISO/IEC/IEEE, Geneva, Des 2016.

39. Zylberman, A., A. Shotten, and Q. Group, "Test language -introduction to keyword driven testing," online article, Software Development Magazine, Geneva, Aug 2010. Available: http://www.methodsandtools.com.

40. ISO/IEC/IEEE, "Iso 14971:2007 medical devices — application of risk management to medical devices," standard, ISO, Geneva, Des 2007.

41. Norweck, J. T, Seibert, J. Anthony, Andriole, K. P, Clunie, D. A, Curran, B. H, Flynn, M. J, Krupinski, Elizabeth, Lieto, R. P, Peck, D. J, Mian, T. A, *et al.*, "Acr–aapm–siim technical standard for electronic practice of medical imaging," *Journal of Digital Imaging*, Vol. 26, pp. 38–52, Feb 2013.

42. Brusan, A., "ibex project repository," online repository, github, Turkey, Feb 2020. Available: https://github.com/altaybrusan/ibex/tree/FastDev.Accessed: 22 Feb 2020.

43. Hall, and G. McLean, *Adaptive Code Via C#: Agile Coding With Design Patterns And SOLID Principles*, USA: Pearson Education, 2014.

44. Gamma, E., *Design Patterns: Elements Of Reusable Object-Oriented Software*, India: Pearson Education India, 1995.

45. Kroll, Per, Kruchten, and Philippe, *The Rational Unified Process Made Easy: A Practitioner's Guide To The RUP*, USA: Addison-Wesley Professional, 2003.

46. NEMA, "Digital x-ray image iod," standard, NEMA organization, Feb 2020. Available: http://dicom.nema.org/dicom/2013/output/chtml/part03/sect_A.26.html Accessed: 22 Feb 2020.

47. Kitware, "Vtk," software, Kitware, Feb 2020. Available: https://vtk.org/download/ Accessed: 22 Feb 2020.

48. computer science institute, O., "Dcmtk," software, OFFIS computer science institute, Feb 2020. Available: https://dicom.offis.de/index.php.en. Accessed: 22 Feb 2020.

49. Toolkit, C. T. C., "Ctk," software, CTK - The Common Toolkit, Feb 2020. Available: http://commontk.org/index.php/Main_Page. Accessed: 22 Feb 2020.

50. Qt, "Qt," software, Qt Framework., Feb 2020. Available: https://www.qt.io/download. Accessed: 22 Feb 2020.

51. Log4Qt, "Log4qt," software, Log4Qt, Feb 2020. Available: http://log4qt.sourceforge.net/. Accessed: 22 Feb 2020.

52. SQLite, "Sqlite," software, SQLite, Feb 2020. Available: https://www.sqlite.org/download.html. Accessed: 22 Feb 2020.

53. Laplante, P., *What Every Engineer Should Know About Software Engineering*, USA: CRC Press, 2007. ISBN 0849372283.

54. Pianykh, O. S., *Digital Imaging And Communications In Medicine (DICOM): A Practical Introduction And Survival Guide*, Springer Science & Business Media, 2009.

55. Herk, M. V., L. Zijp, and J. Meinders, "Conquest dicom software," software, University of Manchester, Feb 2019. Available: https://ingenium.home.xs4all.nl/dicom.html. Accessed: 22 Feb 2020.

56. Herk, M. V., L. Zijp, and J. Meinders, "Dvtk ris emulator," software, DVTk, May 2017. Available: https://www.dvtk.org/dicom/ris-emulator/ Accessed: 22 Feb 2020.

57. Gao, J., H.-S. Tsao, and Y. Wu, *Testing and Quality Assurance For Component-Based Software*, Artech House, 2003.

58. Pittet, S., "An introduction to code coverage," online article, Atlassian, May 2020. Available: https://www.atlassian.com/continuous-delivery/software-testing/code-coverage Accessed: 22 Feb 2020.

59. Cosic, and Domagoj, "An open medical imaging workstation architecture for platform-independent 3-d medical image processing and visualization," *IEEE Transactions On Information Technology In Biomedicine*, Vol. 1, pp. 279–283, Dec 1997.

60. ISO/IEC/IEEE, "Iec 82304-1:2016 - health software — part 1: General requirements for product safety," standard, ISO, Geneva, Oct 2016.

61. IEC, "Medical electrical equipment - part 1-2: General requirements for basic safety and essential performance - collateral standard: Electromagnetic disturbances - requirements and tests," standard, IEC, Geneva, Feb 2014.

62. ISO, "Iso 14971:2019 - medical devices — application of risk management to medical devices," standard, ISO, Geneva, Oct 2019.

63. Ronquillo, J. G, Zuckerman, and D. M, "Software-related recalls of health information technology and other medical devices: Implications for fda regulation of digital health," *The Milbank Quarterly*, Vol. 95, pp. 535–553, Dec 2017.

64. Oglevee, Catherine, Pianykh, and Oleg, "Losing images in digital radiology: more than you think," *Journal of Digital Imaging*, Vol. 28, pp. 264–271, Jun 2015.

65. Grissinger, and Matthew, "Oops, sorry, wrong patient!: A patient verification process is needed everywhere, not just at the bedside," *Pharmacy And Therapeutics*, Vol. 39, p. 535, Aug 2014.

66. NEMA, "Ps3.1 dicom ps3.1 2019e-introduction and overview," standard, NEMA organization, Feb 2020. Available: http://dicom.nema.org/medical/dicom/current/output/pdf/part01.pdf Accessed: 22 Feb 2020.

67. Arcaini, Paolo, Bonfanti, Silvia, Gargantini, Angelo, Mashkoor, Atif, Riccobene, and Elvinia, "Integrating formal methods into medical software development: The asm approach," *Science Of Computer Programming*, Vol. 158, pp. 148–167, Jun 2018.

68. Jetley, Raoul, Iyer, S. Purushothaman, and p. Jones, "A formal methods approach to medical device review," *Computer*, Vol. 39, pp. 61–67, Jun 2006.

69. Brusan, A., "Very short literature survey from supervised learning to surrogate modeling," *Online Archive*, Jun 2012. Available: https://arxiv.org/pdf/1203.4788.

70. Brusan, A., F. A. Durmaz, A. Yaman, and C. Öztürk, "ibex: Modular open-source software for digital radiography," *Journal Of Digital Imaging*, pp. 1–14, Jun 2019.

71. Pressman, R. S., *Software Engineering: A Practitioner's Approach*, Palgrave macmillan, 2005.

72. Blokdyk, G., *IEC 62304 A Complete Guide*, 5STARCooks, 2019.

73. Siewert, B., O. R. Brook, M. M. Mullins, R. L. Eisenberg, and J. B. Kruskal, "Practice policy and quality initiatives: strategies for optimizing staff safety in a radiology department," *Radiographics*, Vol. 33, pp. 245–261, Jan 2013.

74. Doug, R., and S. Matt, "Use case driven object modeling with uml: theory and practice," 2007.

75. Rule, F., "Current good manufacturing practice in manufacturing, packaging, labeling, or holding operations for dietary supplements," *Fed Regist*, Vol. 72, pp. 34752–34958, 2007.

76. Food, D. Administration, *et al.*, "General principles of software validation; final guidance for industry and fda staff," *Food and Drug Administration*, Vol. 11, 2002.

77. Food, H. Drug Administration, *et al.*, "Current good manufacturing practice in manufacturing, packaging, labeling, or holding operations for dietary supplements. final rule.," *Federal Register*, Vol. 72, no. 121, p. 34751, 2007.

78. StickyMinds, "Keyword-driven testing," 2020. Available: https://www.stickyminds.com/article/keyword-driven-testing Accessed: January 13, 2020.

79. of Radiology, A. C., "Acr–aapm–siim-spr practice parameter for digital radiography; acr–aapm–siim-spr practice parameter for digital radiography," *ACR*, Vol. 1, no. 2017, 2017. Available: https://www.acr.org/-/media/ACR/Files/Practice-Parameters/Rad-Digital.pdf Accessed January 12, 2020.

80. of Radiology, A. C., "Acr–aapm–siim practice parameter for electronic medical information privacy and security," *ACR*, Vol. 1, no. 2019, 2019. Available: https://www.acr.org/-/media/ACR/Files/Practice-Parameters/Elec-Info-Privacy.pdf Accessed January 12, 2020.

81. Modi, H. S., N. K. Singh, and H. P. Chauhan, "Comprehensive analysis of software development life cycle models," *Int Res J Eng Technol*, Vol. 4, no. 6, pp. 117–22, 2017.

82. Rausand, M., and A. Høyland, *System Reliability Theory: Models, Statistical Methods, And Applications*, Vol. 396, John Wiley & Sons, 2003.

83. Zielczynski, P., "Traceability from use cases to test cases," 2006. Available: https://www.ibm.com/developerworks/rational/library/04/r-3217/ Accessed: January 29, 2020.

84. Kruchten, P., *The Rational Unified Process: An Introduction*, Addison-Wesley Professional, 2004.

85. Kruchten, P. B., "The 4+ 1 view model of architecture," *IEEE Software*, Vol. 12, no. 6, pp. 42–50, 1995.

86. Bucanek, J., "Model-view-controller pattern," *Learn Objective-C for Java Developers*, pp. 353–402, 2009.