



Assessing Available Open-Source PACS Options

Hamidreza Ghaderi¹ · Alain April¹

Received: 13 June 2020 / Revised: 18 December 2020 / Accepted: 18 February 2021 / Published online: 17 August 2023
© The Author(s) under exclusive licence to Society for Imaging Informatics in Medicine 2023

Abstract

Medical imaging technology is producing a growing number of medical images types as well as patient-related information. The benefits of using modern medical imaging systems in healthcare are undeniable. Picture archiving and communication system (PACS) have revolutionized medical imaging practice. PACS have widely impacted the accessibility of medical images, reduced imaging costs, eliminated the physical storage of films, improved time management of radiologists, and allowed automated decision-making and diagnosis. Many health organizations and manufacturers have invested on developing commercial PACS. However, commercial PACS are not affordable for all hospitals while open-source PACS are increasingly becoming a viable option. Our research project is looking for an open-source PACS for the Donka University hospital of Guinea. Open-source PACS are currently available and are offering varying functionalities, documentation, and technical support from their developer communities. Selecting an open-source PACS is not an easy task and not only depends on the hospital requirements but also requires assessing each open-source PACS to find the best match. In this paper, the most popular open-source PACS are evaluated using a simple comparison approach based on four criteria. The result of this assessment shows that Orthanc, DCM4CHE, DCMTK, and Dicoogle are the most mature open-source PACS according to our criteria and the needs of Donka.

Keywords Medical imaging · Commercial PACS · Open-source PACS · Selection criteria

Introduction

Imaging plays a vital role in modern hospitals and medical imaging research [1]. Managing the growing amounts of medical data produced by hospitals every year (i.e., from terabytes to petabytes) is a current concern for hospital managers [2]. Picture archiving and communication system (PACS) is a technology that helps with managing medical images [3]. It facilitates electronic access to the medical images and allows their storage, transmission, and archiving [4]. Studies report that the use of digital imaging management systems helps hospitals and clinics decrease their overall imaging management costs (such as material costs, physical storage space, and manual labor) as opposed

to using traditional radiology technology [5]. As well, hospitals report that imaging service delivery has also improved because of PACS technology that has eased the imaging workflow, increased the efficiency and productivity of the imaging service, and allowed for time-saving overall [6]. Furthermore, PACS have become the basis for supporting radiologist' decision-making process and providing a better quality diagnosis overall [7].

The development of PACS systems dates back to the 1970s [8] and, over the years, has seen several advancements. The history of PACS development can be described in a number of key evolutionary stages [9]. The first stage of PACS, in the 1970s, saw the development of the initial electronic imaging system repository. In the late 1980s, this initial PACS system integrated with health information systems (HIS) and the radiology information system (RIS) was created. Then, in the early 1990s, the international standard on digital imaging and communication in medicine (DICOM) was published, resulting in standard protocols between medical devices. In the most recent evolutionary stage, the PACS workflow and server application, such as the enterprise PACS and the Web-based PACS, have emerged

✉ Hamidreza Ghaderi
hamidreza.ghaderi.1@ens.etsmtl.ca

Alain April
alain.april@etsmtl.ca

¹ Department of Software and IT Engineering, École de technologie supérieure, A-4467, 1100 Notre-Dame St W, Quebec H3C 1K3 Montreal, Canada

[10]. In the USA, the practical implementation of PACS in hospitals started in the 1980s [8], and very few select hospitals decided to use them [11]. The success demonstrated by this new healthcare technology was followed by the wide adoption of the technology, and PACS were implemented progressively in many hospitals all around the world. For example, they were adopted widely in Asia [12], Europe [13], and North America [12]. Today, a large number of hospitals have PACS. In fact, a hospital in the G8 countries that does not have one is considered a hospital that has not yet understood the value of the technology or cannot afford it. Some developing countries are also beginning to use PACS [14], but the affordability of commercial versions of a PACS prevents a large number of them from acquiring it.

Consequently, the arrival of a mature open-source PACS offering provides a potential solution to this problem for these hospitals all over the world. An open-source version of a PACS provides a foundation for implementing an imaging repository and gradually offering more advanced applications when needed. The goal of developing an open-source PACS is to provide suitable tools that can then be used by software engineers to implement PACS functions in their hospital without the high cost demanded by commercial suppliers [2]. The choice of an open-source PACS solution should be based on certain criteria. There are some popular open-source PACS currently available, and in the next section, four main criteria are defined to evaluate and rank these PACS. The list abbreviations are being demonstrated in Table 1.

Evaluation of Open-Source PACS

This section of the paper presents a number of open-source PACS that are evaluated using four criteria defined as follows: (1) community activities; (2) licensing models; (3) activity, support, and documentation; and (4) enterprise functions and software characteristics. Note that the information provided in the tables below are extracted to the best of our knowledge.

Community Activities

The first evaluation criteria we propose is related to how mature the PACS software project is in its community. Maturity, in this context, refers to its constant development/improvement as well as the level of collaboration and quality control used in the open-source project. Few open-source projects evolve to a high level of maturity, and due to the participation of highly focused developers in collaborative development, open-source projects can compete with and even be superior in quality and functionality, in many cases, to commercial software [15].

Table 1 List of abbreviations

| Abbreviation | Description |
|--------------|--------------------------------------|
| AI | Application integration |
| AU | Activity and utilization |
| CP | Cross-platform |
| EOI | Ease of installation |
| FRD | Final release date |
| IA | Image archiving |
| IC | Image communication |
| ID | Image distribution |
| IM | Image management |
| IP | Image processing |
| IV | Image viewing |
| L | Linux |
| OT | Other |
| PF | Platform |
| PL | Programming language |
| TSF | Technical support forums |
| W | Windows |
| W/L | Windows and Linux |
| WD | Website appearance and documentation |
| WP | Worklist provider |

Forums and shared documentation, often provided via a Wiki, provide a substantial level of customer support for many community-based open-source projects. Using this Wiki, users and contributors can find answers to their questions by reviewing topics reported by others who have faced the same problem and have resolved it or can even create a new topic related to their question. A high level of pertinent forum activity is one of the useful indications of a mature open-source project. Active and vibrant forums provide responsive developers with support from volunteers and code contributors. The last update, or release time, of a software project is another way to judge the maturity of an open-source project. Mature open-source projects update their software monthly, weekly, or in some cases daily [15].

In summary, the contribution of developers and users in an open-source project, the frequency of updates and date of last release, project forum activity, and shared/up-to-date documentation located in a Wiki are good indicators and useful measures to be used to rank open-source PACS. Table 2 summarizes information extracted from the most popular PACS project's source code repositories, which is ranked by considering the number of project forks, final release date, and forum topics count. We can see that DCM4CHE, DCMTK, and Orthanc projects have the highest ranking.

Table 2 Evaluating open-source PACS by developer activity, updating project activity, and community activity

| PACS | Fork | Watch | Star | Release | FRD | Forum topics | Forum members |
|-------------|------|-------|------|---------|---------|--------------|---------------|
| DCM4CHE | 416 | 126 | 620 | 64 | 05/2020 | 5158 | 2543 |
| DCMTK | 117 | 46 | 280 | 42 | 10/2019 | 4717 | 3085 |
| Orthanc | 45 | 21 | - | 24 | 05/2019 | 1949 | 792 |
| ConQuest | 15 | 11 | 33 | - | 04/2013 | 2143 | 107011 |
| ClearCanvas | 400 | 102 | 310 | 3 | 04/2015 | - | - |
| Dicoogle | 77 | 34 | 180 | 7 | 12/2017 | - | - |
| EasyPACS | 35 | 10 | 83 | - | 04/2015 | - | - |
| MRIdb | 11 | 6 | 19 | 2 | 09/2014 | - | - |

Licensing Models

In the process of selecting an open-source PACS for further development, the project license needs to be considered as a very important choice factor. There are several license models for open-source software (OSS), but the Berkeley Software Distribution (BSD) and GNU Project General Public License (GPL) are the most popular licensing models. The BSD license model says you may download a program and use it for non-commercial or commercial products [5]. The main conditions include the following:

- Credit the authors in the source code or reproduce the copyright in binary distributions.
- May not sue the creator, if the software does not work as you think it should.
- May not use the creator's name to endorse the product.

Table 3 presents an overview of the popular open-source license styles used in PACS open-source projects.

The BSD license model is considered a business-friendly license model because hospital can use it, modify it, and even sell the code without paying its authors. The central test note (CTN), which implemented the DICOM international standard, was released with a BSD open-source license. Many companies contribute in open-source software (OSS) projects that use a BSD-style license because it is easier to implement a parallel copy of the source code and continually sync and update these different versions of the source

Table 3 Open-source PACS license

| PACS | License |
|-------------|-----------------------------|
| Orthanc | GPLv3.0 |
| DCMTK | BSD |
| Dicoogle | GPLv3.0 |
| DCM4CHE | MPL v1.1/GPL v2.0/LGPL v2.1 |
| ClearCanvas | GPLv3.0 |
| MRIdb | GPLv3.0 |
| ConQuest | - |
| EasyPACS | - |

code. Collaborating in an OSS project could also have direct benefits in terms of visibility for the companies.

The GPL open-source license model requires that if you release a modified product to the public, the modified source code must be shared with the original creators [16]. This approach helps in that the improvements to the OSS project will circle back and coherently be delivered again to the public. Many developers that create software use a GPL-style license for non-commercial use (e.g., home and academic use). But GPL is more restrictive for commercial use. The GPL-style license allows free use for end-users, but when a company sells the OSS as part of one of their solutions, there is a need for royalties. It means that software is free to use, but if it is sold, some portion of the profits should go to the founders.

The GNU Lesser General Public License (LGPL) allows companies and developers to use and integrate their own source code without the necessity of releasing their own source code. However, there is still a need to make available the modified software component under the LGPL. The letter L for “Lesser” refers to the fact that users can only modify software component under the LGPL, and not the proprietary components.

The Mozilla Public License (MPL) is considered a copy-left license because it encourages contributors to share their modifications with minimal restrictions and allows the inclusion of their source code with source code under other licenses.

Activity, Support, and Documentation

This section assesses the project's activity, support, and documentation using the following perspectives.

- Web site appearance and documentation: current and well-crafted documentation is a good indicator of a successful project. Writing documentation for a software project is often the last task that developers want to spend effort on. In the best open-source projects, users also collaborate with developers in preparing and keeping comprehensive documentation up to date. Documents of a mature open-source project will

Table 4 Evaluating PACS by website appearance and documentation, activity, and utilization, ease of installation, and technical support forum

| PACS | WD | AU | EOI | TSF | PF |
|-------------|-----|-----|-----|-----|-----|
| Orthanc | ••• | ••• | ••• | ••• | CP |
| DCMTK | •• | ••• | ••• | ••• | CP |
| DCM4CHE | •• | ••• | ••• | ••• | CP |
| Dicoogle | ••• | ••• | •• | - | CP |
| ConQuest | • | ••• | •• | ••• | W/L |
| EasyPACS | • | - | • | - | CP |
| MRIdb | • | - | • | • | L |
| ClearCanvas | • | - | • | - | W |

include installation guides, screenshots, user guides, and developer’s guides on a Wiki [15];

- Activity and utilization: statistics provided by the source code repositories such as Sourceforge and GitHub could also indicate a successful project. These Web sites contain some activity measures such as the following: how often the project has been forked or downloaded, when the last update of the code was, the defect reports open and closed, the number of contributors, the number of subscribers to the project, and information about the activity of the bulletin board [15];
- Ease of installation: an easy installation process is another element that could help with the choice of one open-source software over another. Also, whether the software can operate on different platforms (i.e., operating systems), such as Windows, Linux, and Mac OS, is an advantage. Poor installation documentation and insufficient validation tests on different hardware platforms is the most important cause of installation failures. If other users report that an open-source project is hard to install, it could be a sign that the project is not mature enough [15];
- Technical support forums: the existence of an active support forum for an open-source project is a good sign that a large group of active contributors is help-

ing each other to resolve the issues, evolve the software and get the most value from the application. In a mature open-source community, response time when a support request is issued is typically short even for the most challenging questions from volunteers and code contributors [15].

Table 4 shows a quality rating for each activity, support, and documentation characteristic assessed. The three dots (...) in Table 4 indicates good development, resources, high activity, and utilization in recent years, and software that is easy to install. Alternatively, one dot (.) means less development, lower amount of activity and utilization, and software that is harder to install. Finally, a dash (-) indicates that not enough information was publicly available for that characteristic to be assessed. Ease of installation criteria is assessed by averaging documentation quality, platform, and technical support quality. It shows that Orthanc, DCMTK, DCM4CHE, and Dicoogle obtain the best results. These open-source PACS software provide good documentation (e.g., Orthanc book and Dicoogle learning pack) for developers, researchers, and users. Orthanc Google group, DCMTK developer community, DCM4CHE Google group, and ConQuest forum provide a platform for users and developers to discuss with experts related to their issues. These communities help developers and users with the installation process, adaptation, and maintenance.

Enterprise Functions and Software Characteristics

Next, we investigate the different built-in functionality for each open-source PACS, which are typically image archiving, image management, image communication, image processing, image viewing, and image distribution.

Some PACS software operate on limited or specific operating systems, but others can work on many (i.e., they are called cross-platform software). The ease with which programmers can adapt the software to hospital-specific needs is another concern when choosing an open-source PACS. Modular architecture and the availability of development plugins help developers when adapting

Table 5 Open-source enterprise functions and software characteristics

| PACS | IA | IM | IC | IP | IV | ID | WP | AI | OT | PL | Extensibility |
|-------------|----|----|----|----|----|----|----|----|----|---------|------------------------|
| Orthanc | • | • | • | | • | • | • | | • | C++ | API/Library, Scripting |
| DCM4CHE | • | • | • | • | | | • | • | | Java | API/Library |
| Dicoogle | • | • | • | • | • | • | | | | Java | API/Library |
| DCMTK | • | • | • | • | | | | • | • | C/C++ | API/Library |
| ConQuest | • | • | • | • | | | • | | • | C/C++ | API/Library |
| ClearCanvas | • | • | • | • | • | | | | | C/C# | API/Library |
| MRIdb | • | • | | | | | | | • | Java | API/Library |
| EasyPACS | | | | | • | | | | | Js/Java | API/Library |

Table 6 Open-source PACS assessment results using four criteria

| Rank | Criteria 1 | Criteria 3 | Criteria 4 | Result 1 | Result 2 |
|------|-------------|-------------|-------------|-------------|-------------|
| 1 | DCM4CHE | Orthanc | Orthanc | Orthanc | Orthanc |
| 2 | DCMTK | DCMTK | DCM4CHE | DCM4CHE | DCM4CHE |
| 3 | Orthanc | DCM4CHE | Dicoogle | DCMTK | DCMTK |
| 4 | ConQuest | Dicoogle | DCMTK | Dicoogle | Dicoogle |
| 5 | ClearCanvas | ConQuest | ConQuest | MRIdb | ConQuest |
| 6 | Dicoogle | EasyPACS | ClearCanvas | ClearCanvas | MRIdb |
| 7 | EasyPACS | MRIdb | MRIdb | - | EasyPACS |
| 8 | MRIdb | ClearCanvas | EasyPACS | - | ClearCanvas |

the software. A software that provides Restful API is preferred in our assessment, as it facilitates future development. Table 5 considers the number of built-in functions (a dot shows that the open-source PACS includes that built-in function), platform, and extensibility options. Orthanc, DCM4CHE, Dicoogle, and DCMTK PACS have the best results according to these criteria.

Assessment Results

In this research, we have evaluated and compared eight open-source PACS projects using a number of characteristics grouped into four criteria: (1) community activities; (2) licensing models; (3) activity, support, and documentation; and (4) enterprise functions and software characteristics in order to identify what we think are the best open-source PACS to be used in our research project for an African hospital (the Donka University Hospital in Guinea). Table 6 presents a synthesis of this assessment.

In Table 6, the two columns (Result 1 and Result 2) summarize the assessment result. Result 1 is the average for criteria 1, 3, and 4 with BSD or GNU license models, and Result 2 includes all averages without considering the license model.

According to these results, Orthanc, DCM4CHE, DCMTK, and Dicoogle are identified as the top-ranking open-source PACS projects.

Conclusion

This research has presented an open-source PACS selection approach based on four criteria. The result of this assessment shows that Orthanc, DCM4CHE, DCMTK, and Dicoogle are the most mature open-source PACS according to these criteria. We also noticed that both Orthanc and Dicoogle use the DCMTK and DCM4CHE in their internal software

architecture. Orthanc and Dicoogle PACS offer basic functionalities and provide support for future development for developers to customize and enhance their PACS software based on their requirements. In our next publication, we aim to present the implementation approach of Orthanc at the Donka University Hospital located in Guinea as a follow-up to this initial analysis.

References

1. Salvador P, Nogueira A, Gonçalves F: DICOM interception system for independent image backup. *IEEE Network Operations and Management Symposium (NOMS)*, 2014
2. Bui AA, Morioka C, Dionisio JD, Johnson DB, Sinha U, Ardekani S, Taira RK, Aberle DR, El-Saden S, Kangarloo H: openSourcePACS: An Extensible Infrastructure for Medical Image Management. *IEEE Transactions on Information Technology in Biomedicine*, 11:94–109, 2007
3. Law MY, Zhou Z: New direction in PACS education and training. *Computerized Medical Imaging and Graphics*. 27:147–156, 2003
4. Arora D, Mehta Y: Use of picture archiving and communication system for imaging of radiological films in cardiac surgical intensive care unit. *Journal of anaesthesiology, clinical pharmacology*. 30:447–447, 2014
5. Xue Y, Liang H: Understanding PACS Development in Context: The Case of China. *IEEE Transactions on Information Technology in Biomedicine* 11:14–16, 2007
6. Liu BJ, Huang HK: PACS and Medical Imaging Informatics for Filmless Hospitals, *Biomedical Information Technology*, pp. 1–36, (2008)
7. Valente F, Silva LA, Godinho TM, Costa C: Anatomy of an Extensible Open Source PACS. *Journal of Digital Imaging* 29:284–296, 2016
8. Top M: Physicians' Views and Assessments on Picture Archiving and Communication Systems (PACS) in Two Turkish Public Hospitals. *Journal of Medical Systems* 36:3555–3562, 2012
9. van de Wetering R, Batenburg R: Towards a Theory of PACS Deployment: An Integrative PACS Maturity Framework. *Journal of Digital Imaging* 27:337–350, 2014
10. Huang HK: *PACS and Imaging Informatics: Basic Principles and Applications*, 2nd Edition, Wiley-Blackwell, 2011
11. Duerinckx AJ: Equipment Manufacturers' View on PACS" and "The Medical Community's View on PACS. *Journal of Digital Imaging* 16:29–31, 2003
12. Huang HK: Short history of PACS. Part I: USA, *European Journal of Radiology* 78:163–176, 2011

13. Inamura K, Kousaka S, Yamamoto Y, Sukenobu Y, Okura Y, Matsumura Y, Takeda H: PACS development in Asia. *Computerized Medical Imaging and Graphics* 27:121–128, 2003
14. Mendel JB, Schweitzer AS: PACS for the Developing World. *Journal of Global Radiology* 1:1–1, 2015
15. Nagy P: Open Source in Imaging Informatics. *Journal of Digital Imaging* 20:1-10, 2007
16. Hackländer T, Martin J, Kleber K: An Open Source Framework for Modification and Communication of DICOM Objects. *RadioGraphics* 25:1709–1721, 2005

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.