POMLIF
Journal of Library and Information Science
Vol. 4(1) July, 2025

ISSN: 2659-0034
eISSN: 3093-0405

# Advanced Operational Technique and Management of Hardware and Software for Library Automation in Nigeria

**Fadele, O. D.**
The Polytechnic Library, The Polytechnic, Ibadan
olusolad01@gmail.com
+2348138969815

**Adeshina, K.**
The Polytechnic Library, The Polytechnic, Ibadan
skesi@gmail.com

**Sanni, M. M.**
The Polytechnic Library, The Polytechnic, Ibadan
msanni2013@gmail.com

## Abstract

*Information technology has become a game changer, influencing the way libraries and other information centers collect, preserve, and disseminate information to their diverse clientele. This paper explores and offers progressive insights among developmental knowledge of automation in Nigerian libraries. It covers how hardware and software management technique work in advanced library automation in Nigerian libraries, typical hardware and software management in library automation and the maintenance. Operational technique in management of library automation, determinant factors of hardware acquisition in library automation, automation development life cycle in libraries and dimensions of software quality within the library automation were discussed. Conclusion and appropriate recommendations were made for libraries embarking on automation in Nigeria.*

**Keywords:** Hardware, Software, Automation, Technique, Operational, Library

## Introduction

Advanced operational technique simply means a systematic process of making use of current technical methods available for the upliftment of operations among developers (Ayara, 2019). Library automation hardware and software management involves selecting and maintaining the hardware and software that runs the library's automated systems. Engineers organize hardware and software development projects using the advanced operational technique and management of

Hardware and Software principles which is an essential tool for managing the services provided by various libraries in Nigeria.

The library automation system is created to manage, enhance, and oversee all standard operations within the library and any information center. Putting technique and management methodology in place to ensure efficient operations across the project, it will be adequate to have a systematic Hardware and Software development flow to work optimally in projects involving dozens or hundreds of developers, large codebase, and fast changes to code that runs on hardware. Library automation systems employ computer equipment, databases, and software to streamline library operations. These systems can help with tasks like cataloging and metadata management, circulation and patron management, acquisitions, reporting and analytics, and discovery services.

**How Do Hardware and Software Management Technique Work for Library Automation?**

Deverajan (2017) discussed ardware and software management techniques as specific strategies and processes that manufacturers/developers should follow to collaborate efficiently with other developers and the manufacturer of the hardware. The exact rules vary from technique to technique. In general, they contain guidelines that govern how hardware and software code written by individual developers and manufacturers is integrated into a library automaton consisting of hardware and software. For effective library automation, the Library need a robust hardware and software management strategy, focusing on planning, selection, implementation, and ongoing maintenance of both, including Library Management Systems (LMS) like Koha, Dspace , Green Stone and others.

**Typical Hardware and Software Management in Library Automation**

Library automation requires advanced operational techniques and effective management of hardware and software to ensure efficient and seamless operations. Here are some key considerations according to Holland et al (2018):

***Hardware Management:***

1. Server Management: Ensure reliable servers for database storage, backups, and redundancy.
2. Network Infrastructure: Maintain a robust network with adequate bandwidth, switches, and routers.
3. Client Management: Manage public access computers, staff workstations, and mobile devices.

4. Peripherals: Oversee printers, scanners, Keyboard, Hard drive and other equipment.

5. Asset Management: Track and maintain hardware inventory, including barcode scanners and RFID devices.

***Software Management:***

1. Integrated Library Systems (ILS): Manage and update library management software, such as Koha, Folio, Fedora, Dspace, Green Stone or Sierra.

2. Database Management: Ensure database integrity, backups, and optimization.

3. Discovery Layers_: Manage search interfaces and discovery layers, like EBSCOhost or Primo.

4. Digital Collections: Oversee digital repositories and collections, including digital rights management.

5. Security and Access Control: Implement firewalls, antivirus software, and access controls.

**Advanced Operational Technique in Management of Library Automation.**

Colbrook and Smythe (2017) opined that libraries are the most favorite places for book lovers that must ensure some advanced and complete library automation software that is used to make library completely automatic and computerized which are:

1. Automation Workflows: Streamline tasks using scripts, APIs, and automation tools.

2. Data Analytics: Use data to inform decision-making, track usage, and optimize services.

3. Cloud Services: Leverage cloud-based solutions for scalability, flexibility, and cost-effectiveness.

4. Virtualization: Implement virtualized environments for efficient resource utilization.

5. Disaster Recovery: Develop and regularly test disaster recovery plans to ensure business continuity.

6. Staff Training and Development: Provide ongoing training and support for library staff to ensure technical proficiency.

7. User Experience: Continuously monitor and improve the user experience, including website and catalog usability.

8. Collaboration and Partnerships: Foster partnerships with other libraries, institutions, and vendors to share resources and expertise.

Implementing advanced operational techniques in Nigerian libraries operations involves using technology, automation parameters, and data-driven decision-making to improve efficiency, enhance user experience, and manage resources effectively. This includes using library management systems, adopting virtual library systems, and implementing collaborative tools. These techniques also involve continuous improvement and adaptation to technological trends and user needs. After implementing these effectively, Nigerian libraries can optimize their library services to upscale user experience with increase in productivity within Nigerian Libraries

**Determinant Factors of Hardware Acquisition in Library Automation**

In a real sense, hardware purchased should ideally be made in line with the library's overall information technology plan. When hardware is purchased from a single manufacturer or supplier, compatibility is higher than when different manufacturers or suppliers are used. Hardware can be tracked using asset registers. One practical way to perform asset inspections is by placing bar codes on hardware assets and having bar code scanners read them. A good way to make sure all of these hardware components are in good working order is to include them all in a maintenance program that is recorded and to schedule regular maintenance and inspections by qualified experts.

**Typical Example of Software in Library Automation System**

Library software offers comprehensive management for library systems, including modules for material acquisition, cataloging, circulation, serial management, and online public access catalogue (OPAC). Library software, also known as library systems, automation tools, or integrated library systems (ILS).

Ali (2023) described the software's examples in library automation which are:

- Application software: The most popular kind of software is called an application (or app), which is used by the user to accomplish activities. For example, Koha, Folio, Fedora, Dspace, GreenStone, or Sierra, Microsoft Word, Excel, Power Point, Corel Draw; etc.
- System Software: Together with application software, your device's hardware facilitates=the completion of tasks. For example, an operating system, compiler, or translator

- Middleware: The role of middleware software is to facilitate the switching between application and system software. Examples include message-based communication, data management, and authentication.
- Driver software: Integrated within the system software, it aids in the management of external devices connected to your computer.
- Closed software or proprietary software: closed-source software, or proprietary
- Utility software: This kind of system software can assist you in controlling program performance and keeping your gadget in good working order.
- Open-source software: This category of software permits users to modify the code and distribute it to other users. Any user can utilize these apps that you provide with them for a number of purposes. The following are a few instances of open-source software: Web browsers like Koha, LibreOffice, Firefox, and Greenstone

**Software Development Life Cycle**

The term "Software Development Life Cycle" describes a process-oriented methodology for producing high-quality software. The Software technique, in detail, concentrates on the following stages of software development:

• Planning;
• Software design, including architectural design
• Software development
• Testing and Deployment.

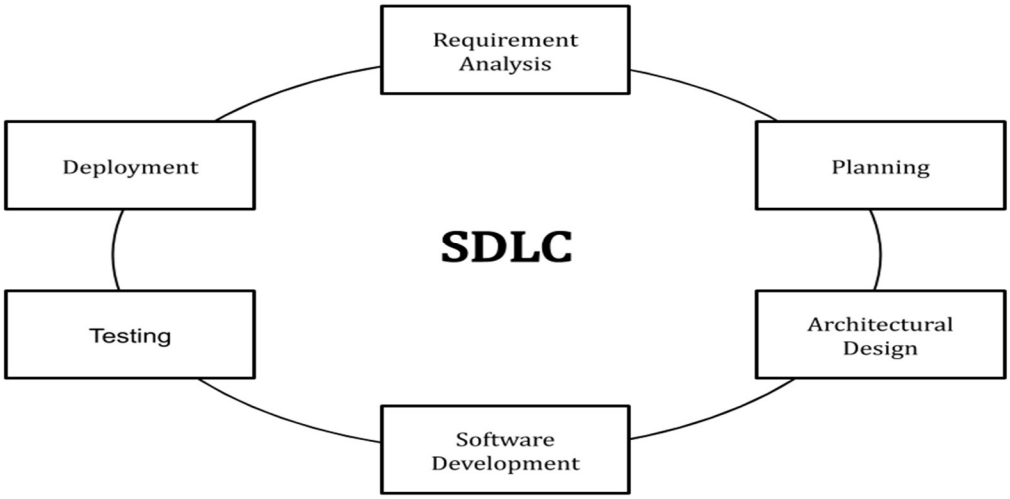
**System Development Life Cycle (SDLC) of Library Automation Software**

According to Aswal (2010) Software Development Life Cycle (SDLC) operates by reducing software development expenses while also enhancing quality and decreasing production time. SDLC apparently achieves these goals by following plans that removes the typical pitfalls of software development projects.

That plan starts by evaluating existing systems for deficiencies, defines the requirements of the new system and then creates the software through the stages of analysis, planning, design,

development, testing, and deployment with anticipating costly mistakes like failing to ask the end-users or client for feedback, SLDC can eliminate redundant rework and after-the-fact fixes.

**System / Software Development Life Cycle**




***Source:*** ***https://www.library-automation-training.com/user's-manual***

**Software Development Techniques and Management in Library Automation**

| Strategy | Benefits | Drawbacks |
|---|---|---|
| Waterfall | One of the simplest development techniques | Infrequent releases can lead to delays |
| **Agile** | Prioritizes rapid application updates | Risk that constant small changes will fail to add up to meaningful large change; requires close coordination among developers |
| DevOps and CI/CD | "Continuous" application updates; addresses the entire application lifecycle | Can require extensive communication and change management |
| Lean Development | Focuses on efficiency, minimizing wasted effort and wasted code | Requires close coordination among developers |

***Source:*** ***https://www.system-analysis-and-design-for-librarians/automation***

The dimensions of software quality include the following characteristics:

Kaspar et al. (2021) described the following aspects of software quality as outlined below: **Accessibility**: How well software can accommodate a wide range of users, including those who need assistive technology like voice recognition and screen magnifiers.

**Compatibility**: The capacity of the program to operate in a range of settings, including with various operating systems, hardware, and web browsers.

**Efficiency**: The program's capacity to function well without squandering resources, money, time, energy, or effort.

• Functionality: The capacity of software to do its designated tasks.

• Compatibility: The program's installation suitability in a certain setting.

**Localization**: The range of languages, time zones, and other characteristics that a software is capable of operating in.

**Maintainability**: The degree to which the program may be changed to incorporate new features, address bugs, etc.

**Dependability**: The capacity of the software to carry out a necessary task under particular circumstances for a predetermined amount of time without making mistakes

**Capability to Grow**: The degree to which the program can adapt its performance to alter as its processing requirements change.

**Safety**: The software's defenses against data loss, theft, illegal access, privacy invasion, and dangerous malware, among other things.

**Capability to Test**. How simple it is to run software tests.

**Practicality**: How user-friendly the program is.

**Performance:** How quickly the program operates when subjected to a given load.

**Portability**: The program's capacity to be quickly moved from one place to another.

**Maintenance of Hardware and Software in Library Automation**

After you have downloaded or purchased something, your physical device or software may occasionally require maintenance. Updating the application or fixing any issues it may run into are examples of maintenance for your software. You can use your computer and the software settings to check for updates or faults in the program. You can also update your program by applying software patches. Programs that have software patches placed on them can assist fix bugs and

update when needed. Software patches can be obtained from the websites of developers and manufacturers (Onyemaizu, Omoike & Ogunjobi, 2019).

**Conclusion**

Almost all hardware and software development initiatives require a development technique and management strategy in place to enable efficient operations with very small and simple projects. The optimal methodology for a particular project of library automation relies on variables including the team's importance of quality and efficiency and how soon hardware and software need to be changed. However, the fundamental advantage of these methods is that they aid in preventing confusion and needless work in intricate automation projects and routine procedures of the library related to hardware and software development acquisition.

**Recommendations**

1. For libraries that are planning to computerize/automate their services, they are to ensure that appropriate hardware and software are selected right from the onset.
2. Considering the importance of library automation in this present age, it is recommended that library automation should be given more priority in terms of funding and appropriate expert should be on ground.
3. Libraries should redesign their automation programme from traditional method to mechanized one especially in this era of artificial intelligence that actually dominate to impact more on students' academic activities.
4. Training and re-training is needed for automation librarian/system analyst or whosoever is in-charge to be abreast of latest development on hardware and software needed to suite the purpose of the system on ground and avoid obsolete architecture / design.

**References**


Alamitos, California, (2007). Is Preventive operational maintenance Known to Uks, Chapin N? In the IEEE Computer Society Press proceedings of the International Conference on Software Performance. Los.

Ali, A. (2023). Software metrics in library automation in new era. New Delhi: Rasmed Publications.

Almatos, C, A, (2018). IEEE Computer Society Press, Los Alamitos, CA, pp. 247–249.

Aswal, R.S. (2010). Library automation for 21st century. New Delhi: S.S.Publications.

Ayara, A. (2019). Advanced operational technique for automation librarians in a Nigerian university library. *African Journal of Library, Archives and Information Science* 10.1: 63.

Biscaglia, V, Malaguti M, Gualandi, P.Boehm & Brown, B. W. (1982). Management planning on mv distribution network, CIRED 63, 2–5 June 1982, Conference Publication No. 438.

Briand, L. C, (1993). Measuring and evaluating maintainability at the conclusion of high level design. IEEE Computer Society Press, Los Alamitos, CA, Proceedings for International Conference on Software Maintenance.

Biscaglia V, Malaguti M, Gualandi P.Boehm, B, W, and Brown, B. W. (2010). Management planning on MV Distribution Network, CIRED 63, 2–5, Conference Publication No. 438.

Charpin, N, Hale J.T. E, Khan, K. M, & Rami1, J. F (2018). Types of software evolution and advance software maintenance. *Journal of Software Maintenance*, 10 ( 4): 5.

Colbrook, A & Smythe C, (2017). The Retrospective introduction of abstraction into software. Proceedings for International Conference on Software Maintenance, IEEE Computer Society Press.

Deverajan, G, (2017). Information Technology in Libraries: Software packages, New Delhi: Ess Publications.

Holland, B., Browd, B. W.& Kasphar JR. (2018). Features of a good software quality, Kcl version of Software Technology, Amsterdam, North.

Kaspar JR, et al., (2021). Features of software quality, TRW series of software technology. Amsterdam, North Holland.

Onyemaizu, C. O., Omoike, D. A., and Ogunjobi, T. E. (2019). Maintenance of hardware and software in library automation arena in of library services in Southwest Nigeria: A Case study of KDL and Unilag libraries. *The Information Technologist* 9.1: 124-125.