

Developing an integrated library management system with agile methods, the University of Ibadan experience

Ihuoma Sandra Babatope

Head Readers' Services Unit

College Library, Delta State College of Education, Mosogar, Nigeria.

Ihuoma.babatope@descoem.edu.ng

Oluwatosin Fisayo Idowu

The Polytechnic Ibadan Library, Sango, Polyroad, Ibadan, Nigeria

fikkyoye@gmail.com

Patrick Uzodinma Agwu

Head, Technical, Infrastructure and Digital Services Unit

Kenneth Dike Library, University of Ibadan, Ibadan, Nigeria.

pu.agwu@mail.ui.edu.ng

Rita Dumbiri

Head Technical Service Unit

College Library, Delta State College of Education, Mosogar, Nigeria.

rita.dumbiri@descoem.edu.ng



Copyright © 2022 by Ihuoma Sandra Babatope, Oluwatosin Fisayo Idowu, Patrick Uzodinma Agwu and Rita Dumbiri. This work is made available under the terms of the Creative Commons Attribution 4.0 International License:

<http://creativecommons.org/licenses/by/4.0>

Abstract:

Agile methods is the software development methodology and practices used in software industry in which the elements and principles of agile methods are used. Generally, the agile development process is an incremental work process that promotes the importance of customer satisfaction, collaboration, communication, teamwork, good quality and planned follow-up practices. The paper aims to discuss these issues. This paper first presents an overview of agile methods, addresses the most relevant ones for developing library management system then discusses the utilization of these methods in the context of the development of an integrated library management system presenting the case of Kenneth Dike Library, University of Ibadan. The Kenneth Dike Library, University of Ibadan after several attempts at automating its processes since the early 90's eventually resolved to develop an in-house Library Automation System in 2015 using agile methods. The paper refers to definite initiatives taken to improve development processes as well as discuss the benefits and challenges when applying the agile methods in order to change traditional working culture and encourage end user participation in the building process. The following elements of agility can be considered the most effective and relevant: appreciating the needs of the customers, end-user participation and satisfaction, quality assurance, ability to redesign and make decisions fast and empowering the team, eliminating of waste. The paper

finally highlights what the library can achieve by adapting a fresh approach for developing, evaluating and managing its operations and how the library staff can benefit from the agile way of working.

Keywords: Automation, Agile Development methods, Management, Quality, User satisfaction and University libraries

Introduction

An obligatory responsibility for librarians and libraries now is to develop and keep up with the emerging trends in technologies in order to satisfy the ever changing needs of all the categories of library users. To achieve this, it requires flexibility, developing an attitude of change, resilience, trainings to keep up with the happenings in the LIS world, collaboration, adaptability to new technologies and creating unique ways in providing value added services to survive the future in libraries. The future is indeed challenging and at the same time exciting for all libraries hence they must be actively engaged to provide services that meaningfully support teaching, learning and research. According to Mathews in Stoddard et.al (2019) “Futurism warns us to avoid tunnel vision and instead to keep an agile mind open to many alternative futures and to adapt our organizations to thrive in whatever conditions arise”, building capacity for adaptation into any organization is a process, it takes time and it requires working with staff who are accustomed to longstanding traditions of “how we do thing in libraries.”

Recently most libraries are developing puissance and the practical know-how to build their collections and have remained resilient notwithstanding the low budgetary allocation, lack of skill training for staff for library automation, keeping up with the emerging technological trends in the face of constant changes, poor attitudinal behavior of the institution management and to mention but a few. This therefore has become a challenge for libraries and librarians to make efforts to address these concerns by trying a novel approach in managing and developing its collections in an organized and standardized manner which is cheaper, faster and easy to retrieve.

Lewis (2016) in Reimagining the Academic Library, has written that “a cultural change in the library will be required and librarians will need to think of themselves as part of a team, not as individual and isolated contributors. This paper presents a case study about how Kenneth Dike Library (KDL), University of Ibadan has attempted to address these concerns by trying a new Agile approach to library automation software development.

Agile Software Development

Abrahamsson, et al (2002) denotes Agile as “the quality of being agile, ready for motion, responsive and dexterity in motion”. Agile Software development can be seen as striping away much heaviness, commonly associated with traditional software-development methodologies, to possibly promote quick response to changing environments, changes in user requirements, accelerated project deadlines, and the like as well (Erickson, Lyytinen & Siau, 2005 pg.89).

Brush and Silvertone (2019) defines Agile software development as a type of development methodology that anticipates the need for flexibility and with a level of practical approach to the delivery of the finished product, which requires a cultural reposition in many companies because it focuses on the clean delivery of individual pieces or parts of the software and not on the entire application. Another author Collier (2011) defined it as practices that include

requirements discovery and solutions improvement through the collaborative effort of self-organizing and cross-functional teams with their customer(s)/end user(s).

Brief History of Agile Software Development

By the early 1990s, a small group of software industry leaders had started developing and promoting new approaches to the Software Development Life Cycle (SDLC) that focused on reacting and adapting to all changing requirements and technologies. Rapid Application Development (RAD), Scrum, extreme programming and Rational Unified Process (RUP) arose at this time as the new, flexible and highly responsive development methods (James, 1991). In 2001, seventeen software developers met at a resort in Snowbird, Utah to discuss lightweight development methods on emerging technologies. It was there the term Agile software development was first used to describe flexible software development that occurred in iterative stages; it became the blanket term for the new methodologies. In an attempt to distinguish Agile software development from traditional methodologies, the group of industry leaders defined a set of values for using Agile and created the Agile Manifesto which was published as Manifesto for Agile Software Development (Beck et.al, 2001). According to Brush & Silvertone (2019), the four core values outlined in the Agile Manifesto are as follows:

Individual interactions are more important than processes and tools: People are the driving force for the development process and respond to business needs, hence they are the most important part of development and should be valued above processes and tools in order to respond to changes and meet customer needs.

A focus on working software rather than thorough documentation: Agile does not eliminate the use of documentation, rather it streamlines it in a way that provides the developer with only the information that is required to do the work -- such as user stories and places value on the process of documentation but higher value on working software.

Collaboration instead of contract negotiations: Agile focuses on collaboration between the customer and project manager rather than negotiations between the two, to work out the details of delivery, thus making it easier for teams to meet the needs of their customers.

A focus on responding to change: The short iterations in the Agile cycle allow changes to easily be made, as it believes change is always a way to improve the project and provide additional value.

According to Beck et al. (2001), Agile Manifesto also outlined 12 core principles for the development process. They are:

1. Satisfaction of Customers early and continuous delivery of valuable software.
2. Welcoming changing requirements, even in late development.
3. Frequent delivery of working software (weeks rather than months)
4. Close, daily cooperation between business people and developers
5. Building projects around motivated individuals who can be trusted
6. Face-to-face conversation is the best form of communication (co-location)
7. Working software is the primary measure of progress
8. Sustainable development and ability to maintain a constant pace
9. Continuous attention to technical excellence and good design
10. Simplicity—the art of maximizing the amount of work not done—is vital
11. Best architectures, requirements and designs emerge from self-organizing teams

12. Regularly, the team reflects on how to become more effective, and adjusts accordingly

Agile software development methods support a broad range of the software development life cycle (Abrahamson, 2002). The Agile software development life cycle is the organized and configured series of phases that a program passes through from beginning to the end. Agile software development method works by breaking the program into small bits of user functionality (user stories), prioritizing them, delivering them steadily in a short time cycle, usually called iterations and each iteration is considered as a short time "frame"(Tyagi, 2020).



Source: Agile Software Development Life Cycle (analyticssteps.com)

The following are the phases in the Agile Software Development Life Cycle (Tyagi, 2020)

1. **Concept:** In this phase, requirements like the objectives, manpower, period of development, benefits and the efforts needed to build the project are clearly defined. Based on this information, the financial implication is ascertained.
2. **Inception:** This phase acts as the planning stage for the duration of the project, beginning from narrowing down the agile framework to be used, to highlighting the features, functionalities, development workflow, how the end product will look like and more relevant requirements from the application.
3. **Design Development and Construction:** Here the actual development process is launched to enable designers, programmers and developers commence their project.
4. **Testing/ Integration:** In this phase, the quality assurance team examines the product's performance to identify and look out for possible bugs or defects from the project.
5. **Implementation:** This phase is when the application is tested and ready for implementation, here the developers can either release the beta version of the application or the final version if they are self-assured about the application.
6. **Retirement /Feedback:** This last stage of Agile SDLC is concerned with feedback, where the feedback received about the product is worked on, as the applications undergo minor updates and changes.

Agile Project Management (APM)

Muslihat (2018) explained APM as a methodology used to deliver complex projects due to its adaptiveness. APM is built on collaboration, flexibility, continuous improvement and attainment of high-quality results. In the same view Etim, (2020) reinforced that the APM process ensures a high success rate when used to manage projects. The high success rate of APM is attained by dividing projects into sections, obtaining feedback and accepting collaborations from both stakeholders and customers to develop strategies that enhance output either at the unveiling of new projects or at each stage of the project.

There are various frameworks used for development, delivery or service provision of APM. Each of these frameworks has specific approaches and outcomes. The outcome of the project will determine the agile approach to be selected and applied. Although, these approach have their own set of characteristics and terminology, they still share common principles and practices. Two common ones that support Agile development are SCRUM and KANBAN

SCRUM

Scrum was created by Jeff Sutherland and Ken Schwaber who were also part of the 17 founders of the Agile manifesto. Scrum is a framework that implements the ideas of Agile software development. It has five values: commitment, courage, focus, openness and respect. It also has goals: develop, deliver, and sustain complex products through collaboration, accountability and iterative progress (Muslihat, 2018). Griffiths (2012) noted that scrum can be distinguished based on a set of activities: roles, events and artifacts.

Scrum Roles is divided into three:

- Product owner: consists of experts who represent the stakeholder and control the product backlog
- Development team: they are the professional that delivers the product (developers, programmers, designers and testers).
- Scrum master: are controllers that ensure that the process and implementation of scrum is followed

Scrum Event:

- Sprint: is the heart of scum. It is a timebox for specific duration for potential project to be completed. The time frame does not exceed a month and is consistent throughout its development phase.
- Sprint planning: this allows room for proper planning at the beginning of every sprint.
- Daily scrum: it's a 15minute time-boxed meeting held every day of the sprint.
- Sprint review: it's an informal meeting held at the end of sprint where the team present their increment to the stakeholder for discussion and feedback.
- Sprint retrospective: a meeting held to reflect on previous sprint and establishes improvement for the next sprint.

Scrum Artifact

- Product backlog: it contains an ordered list of requirements to be developed in a future sprint. It is usually managed by the product owner
- Sprint backlog: it contains requirements and tasks that must be accomplished in the current live sprint.

- The product increment: it is the total product backlog that was completed during a sprint and it is also combined with all previous increments.

Alexander (2018) emphasized that Scrum's focus is particularly on managing software development using the iterative approach. The research also shows that Scrum synergizes all its processes i.e. from sprint to sprint backlog. It also ensures that all levels of stakeholders are involved in each sprint to permit interaction and room for feedback. Schwaber and Suther (2017) noted that Scrum was not the only developed software industry that can make use of scrum but for research and identifying viable markets, developing products, developing and sustaining cloud infrastructure and services among others which is what made it possible for libraries to access or use the software.

KANBAN

The Kanban Method was defined by David Anderson in the early -to-mid 2000s, in response to some of the challenges of the various Agile methods, especially Scrum (Muslihat, 2018). These methods, while trying to solve the challenges of traditional/ waterfall methods, became victim to some of the same challenges themselves. Barolli, Nishino and Miwa (2019) stated that Kanban means "Just in time" that is doing what is needed, when it is needed and at the needed or adequate amount. The Kanban concept is mainly described through visualization and visual management. It paints a picture of the workflow process and also aims to identify any difficulty that might occur or be experienced earlier on in the process in order to ensure high product and service delivery. Griffiths (2012) identifies some principles that support Kanban behavior in a project and it includes:

- Visual workflow
- Limit work-in-progress
- Manage flow
- Making policies explicit
- Using feedback loops
- Improve collaboration/ experimental evolution

Kaban achieves efficiency through visual cues to signal certain stages of development process. The cues include: Kanban board, Kanban Cards and sometimes Kanban swimlanes. Kanban board are visual tools used to visualize development process. It can be the use of whiteboards or virtual board which can be used for personal or professional productivity; Kanban cards are used to communicate progress and represent information while Kanban swimlanes are visual items on the board used to differentiate tasks or items by categorizing them.

In nowadays academic libraries, the deployment and utilization of applications software such as library software packages have become popular and Library software packages are designed primarily to carry out library functions such as charging and discharging of books, library statistics compilation, acquisitions, serials control, bibliographic control, overdue compilation, cataloguing, and so on (Madu, 2004). According to Ifidon and Okoli (2002), some of the outstanding benefits accrued over time through the deployment of library software in the university library include:

Increased speed of myriads of library operations

Improved performance of library staff

Improved storage and retrieval of information

Creation new services such as current awareness services, recent accessions list, access to

online databases, Selective dissemination of Information (SDI) and resource sharing.
Improved statistical records and recording

Breeding (2012) stated relative features of the Integrated Library system as: “provision of computer automation for varied library operations such as cataloguing for creating bibliographic records that represent work in the library collection, circulation that automate works pertaining to loaning of materials to patrons, serial for managing periodical and serial acquisitions for the procurement of new materials for the library collection and OPAC to allow library prospective users to browse through library collections”.

According to Ukachi, Nwachukwu and Onuoha (2014), library software comes in two divisions; Proprietary software – (requires subscription fee) and the Open sources software – (requires no subscription fee). Majority of the libraries in developing countries particularly Nigeria has adopted Open source software because of its reliability, effectiveness and affordability. It is also more flexible in usage and accessibility than those licensed restriction software.

The University of Ibadan Experience

Kenneth Dike Library was originally known as University of Ibadan library and precisely in 1988 it was renamed Kenneth Dike Library in honour of the first African Vice Chancellor Professor Kenneth Onwuka Dike. Kenneth Dike Library (KDL) was established in 1948 and it is one of the biggest University Libraries in West Africa. KDL majorly operates centralized library system whereby the management, acquisition and cataloguing processes are done in the central library. The University of Ibadan usually maintains a library system with Kenneth Dike Library as the central hub of the system. Other libraries in the system include Latunde Odeku Medical Library, 26 faculties, departmental and institute libraries.

In pursuit of the global shift from manual to automated library system, KDL embarked on automation with library management system (Kingdom & Baro 2014). According to Ola (2010) Kenneth Dike library adopted CDS/ISIS software that was developed and distributed freely by United Nations Educational, Scientific and Cultural Organisations (UNESCO) in 1993. With this software, KDL computerized its cataloguing records. Though the software was very flexible and customizable, but it became necessary to adopt another software as their records increased. Then KDL migrated to Tinlib which was introduced by National Universities Commission (NUC) and funded by World Bank Project in 1994. This software lacked basic elements of flexibility (it was DOS – Based). Hence the library adopted the software known as Alice for windows in 2004. This software was not Marc 21 compliant even though it was window based.

Visionary Technology for Library Solutions (VTLS) was equally signed up in 2008 with the approval of university administration in collaboration with 5 other federal universities with the support of Carnegie and Mac Authur Foundation to adopt VIRTUA Integrated Library system software. Ola (2010) highlighted features of VIRTUA as:

1. Internet based and highly flexible software
2. User friendly, time saving and customizable software package
3. Possessing facility for automatic Selective Dissemination of Information (SDI)
4. Supporting virile networking and resource sharing programme
5. Providing facility for Inter- Library Loan (ILL) system among network libraries.

The users of VIRTUA found it to be flexible because it is highly versatile and it is internet based. It is Marc 21 compliant hence it has interface for sharing resources, user friendly and time saving.

Kenneth Dike library had to discontinue the usage of VIRTUA because of the humongous cost and the expiration of the MacArthur Foundation Grant plus the fact that there was a proprietary change accompanied with several operational amendments that hindered smooth handshake between KDL Servers and the new management of the VIRTUA software.

Presently, the library is using locally designed library software known as University of Ibadan Library Integrated Software. It has been discovered that automation process in most academic libraries have failed woefully in the absence of adequate feasibility studies and continuous search for appropriate software by individual libraries (Kingdom and Baro 2014) . In most cases, the academic libraries depend almost absolutely on open source / free software. According to Imo and Igbo (2011) the problems associated with libraries software are: paucity of funds for university libraries to acquire appropriate software, lack of maintenance support, lack of user integration and consideration at design stage and lack of trained staff to manage the software.

Developmental Approach

Agile method was adopted in the development of the UI-ILS emphasizing incremental delivery, team collaboration, continual planning, continual learning, instead of trying to deliver it all at once near the end. The development was done in modules, learning from past challenges and involving the end users at every stage of the developmental process. The overall software development task was broken into parts as individual short term project that eventually formed part of the long term development task. Stakeholders and users of each individual module/units were brought together for requirements gathering, designing of the requirements using a workflow, the designers began the design (Construction), Testing/ Quality assurance was done using committee system before deployment after which Feedback was received and perfected before moving to the next module

The University of Ibadan Integrated Library Software is developed to effectively manage the basic housekeeping functions of the library and it is currently being deployed in phases. It is a multi-tier, client-server architecture that is broad, stable and includes what libraries need to meet most of their technological challenges. UI-ILS modules support simple day to day library transactions, from cataloguing and classification to placing orders for items to acquisitions to charging and discharging and so on, with an unparalleled technical processing features and workflow efficiencies which offer librarians real time information and complete functionalities to carry out their tasks.

The modules implemented so far include: Acquisitions, Cataloguing, Circulation, Serials, OPAC and Patron modules.

Modules

Acquisitions Module

Being the entry point for most resources of the library, proper documentation of the library records starts here. It leverages on other modules like cataloging, circulation etc. This Module allows for:

- Ordering of Items and resources from vendors
- Registration of acquired items
- Managing of vendors and donors
- Managing of item requests from patrons
- Managing repurchase of lost items
- Acquisition reports, etc.

Cataloguing Module

Items are classified and catalogued in this module based on chosen standards. Once items have been recorded in the acquisition register, it moves to this module automatically for the next phase. It leverages on modules like acquisitions, amongst others. This module allows for:

- Cataloguing of items via cataloguing templates or via remote databases from the Library of Congress, Hathi Trust, Open Library etc.
- Display of acquisitions waiting to be cataloged
- Managing cataloguing templates using selected marc fields and marc subfields.
- Managing of item classes.
- Managing of digital entries.
- Cataloguing reports.
- Managing of Marc fields and subfields.
- Online Public Access Catalogue (OPAC). The cataloguing process involves

Serials Module

This module leverages on the excellent features of the cataloguing module as serials eventually will need to be catalogued. It also allows for:

- Managing serials subscriptions.
- Managing issues of serials.
- Managing of articles in issues
- Cataloging of items via cataloging templates or via remote databases from the Library of Congress, Hathi Trust, Open Library and so on.
- Display of serials waiting to be cataloged.

Institutional Repository Module

It also allows for:

- Self-submission and upload of digital copy of works by original authors
- Acceptance workflow by catalogers
- Cataloging of metadata of digital file
- Seamless integration into the OPAC
- Patron request workflow for access to digital copies
- Public request workflow for access to digital copies
- Approval system integrated with circulation system

Circulation Module

This module helps to a large extent to simplify the complexities in the circulation system to simple processes and it interfaces with all modules from the patrons to the cataloguing and even the acquisitions. It allows for:

- Managing of patrons.
- Check out and check in of Items (with option of barcodes).
- Managing of patron types (including item borrow limit, borrowing duration and renewal limits).

- Managing of loan types (including borrowing duration and lateness fines).
- Managing of library shelves.
- Managing item reservations.
- Managing overdue and lost items
- Managing fines.
- Item Recall.

Patron Module

Patrons are the primary reasons why libraries are created and as such service delivery at this point is highly optimized. With each patron having an account, they are able to use the Library in a somewhat self-service mode. This module allows for:

- Advanced searches via the OPAC (Online public access catalog).
- Item submission for Institutional Repository (IR)
- Reservation requests for library items.
- Renewal of checked out items.
- Item status and locations.
- Request for new Items.
- Virtual shelves.

Admin Module

This Module caters for the general preferences of the software. An engine room for self-maintenance and lends support to other modules. It allows for:

- Creating of administrative users (catalogers, acquisition users, and so on).
- Manages data back-up.
- Manages library locations.
- Manages data migration.
- Administrative reports (circulation reports, acquisition reports, shelf listings, patron reports, reservation and request reports).

Benefits of the In-House Library Software Application to Library Processes

- Unlimited number of users for all modules.
- Multi library location: administrative users from different Library Locations can use the software over the network.
- Easy reporting: reports are easily generated and produced as pdf.
- Enhanced communication interfaces for delivery of mails and notifications to library users.
- Self-service for library patrons.
- Quick and flexible back up.
- Unlimited databases.
- All-round customer support.
- Software updates.
 - Guaranteed return on investment.

Pitfalls of the In-House Library Software Application to Library Processes

- Cost of maintaining power and other hardware infrastructure is very high
- Sustainability of technical personnel
- Poor attitudinal behavior towards the system by some library staff
- Lack of sponsorship to attend skilled training programmes that can boost staff performance

Conclusion

This paper presents an overview of agile methods, addressing the agile values, principles, agile project management and agile software development processes. It highlighted the characteristics and features of agile software development approach and identified the most relevant ones for developing library management system, then discusses the utilization of these methods in the context of the development of an integrated library management system in Kenneth Dike Library, University of Ibadan. This study investigated and explored the historical perspective of automation efforts of the University of Ibadan Library from inception to the final state of developing an in-house integrated library management software. It further discussed the implementation of Agile methods in requirement gathering, design, development, testing and deployment of the in-house Integrated Library Management Software. However, the major thrust of this present study showcases benefits of agility in the development and utilization of Integrated Library Software as well as the pitfalls. It also highlights the components and features of varied modules such as acquisition, cataloguing and circulation. It has been discovered during the course of the study that deploying Agile methods in library software development encouraged end user acceptability and use of the software.

Acknowledgments

Appreciation goes to the Tertiary Education Trust Fund (TETFUND) Academic Staff Training and Development Intervention Scheme for Conference Attendance sponsorship, which made it possible for this paper to be presented at the 2022 IFLA WLIC Conference in Dublin, Ireland. Also gratitude goes to the University Librarian of Kenneth Dike Library (KDL), University of Ibadan and the management team for their support during this research.

References

- Abrahamsson, P., Salo, O., Ronkainen, J. & Warsta, J. (2002). Agile software development methods: Review and analysis, VTT publication 478, Espoo, Finland, 107p.
- Breeding, M. (2012). Open Source Integrated Library System. ALA Tech Source. Retrieved from <http://www.alatechsource.org/tr/open-source-integrated-library-systems>. Chisenga, I. (2006). Global information and libraries in Sub – Sahara Africa. *Library management* 21(4): 174.
- Brush, k. and Silverthorne, V. (2019). agile software development. *Techtarget* <https://www.techtarget.com/searchsoftwarequality/definition/agile-software-developmentby-information-systems-professionals>.” *iee transactions on engineering management* 47(3).
- Erickson, J., Lyytinen, K., & Siau, K. (2005). Agile modeling, agile software development, and extreme programming: The state of research. *The Journal of Database Management*, 16(4), 88-100. <http://www.doi.org/10.4018/jdm.2005100105>
- Etim, S. A., Jaiswal, P. C., Subroto, M. & Ortege, C. F. V. (2020). Managing information technology project using agile methodology: The case of book for Africa project. In S.A. Etim (Ed) *Developing countries & technology inclusion in the 21st century information society: advances in IT standards & standardization research* (pp 123 - 150). IGI Global for agile software development,” 2001.
- Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick,

- Griffiths, M. (2012). PMI-ACP Exam Prep: RMC Publications, Inc.
- Hussain, S. and Ansari, M. A. (2009). Library Automation softwares packages in India: a study of the catalog modules of Alice for windows, Libsys and virtua.’’. *Annals of Library and Information Studies*, 54(2): 146-148.
- Ibrahim, N. (2007). An overview of agile software development methodology and its relevance to software engineering. *Jurnal Sistem Informasi*, 2(1), 69 - 80
- Ifidon, S. E. and Okoli, G. N. (2002). Forty years of academic and research library services in Nigeria, past, present and future, in Nigeria Library Association – 40th Annual National Conference and AGM. A Compendium of papers presented June 16th – 21st 2002.
- James, M. (1991). Rapid Application Development. Macmillan. ISBN 978-0-02-376775-3.
- Ken W. Collier, (2011). *Agile Analytics: A Value-Driven Approach to Business Intelligence and Data Warehousing*. Pearson Education. pp. 121
- Kent Beck; James Grenning; Robert C. Martin; Mike Beedle; Jim Highsmith; Steve Mellor; Arie van Bennekum; Andrew Hunt; Ken Schwaber; Alistair Cockburn; Ron Jeffries; Jeff Sutherland; Ward Cunningham; Jon Kern; Dave Thomas; Martin Fowler; Brian Marick (2001). "Manifesto for Agile Software Development". Agile Alliance. Retrieved 14 June 2010.
- Kingdom H. K. and Ebikabowe Emmanuel Baro (2014). The use of Library software in Nigeria University Libraries and challenges. *Library Hi –Tech News* (3):15-17
- Lewis, D. (2016). Reimagining the academic library. Lanham, MD: Rowman & Littlefield.
- Lowell, L. & Ron, J. (2004). Extreme Programming and Agile Software Development Methodologies’, *Information Systems Management*, vol. 21, pp. 41-61.
- Madu, E. C. (2004). Automation and service provision in Libraries and information centres in developing countries in Madu, E. C. (ed) *Technology for information management and services: modern libraries and information centres in developing countries*, Ibadan: Evi- Coleman Publications.
- Mathews, B. (2014). Librarian as futurist: Changing the way libraries think about the future. *Portal: Libraries and the Academy*, 14(3), 453–462. doi:10.1353/ pla.2014.0019
- Morgan M. Stoddard, Bill Gillis & Peter Cohn (2019) Agile Project Management in Libraries: Creating Collaborative, Resilient, Responsive Organizations, *Journal of Library Administration*, 59:5, 492-511, DOI: 10.1080/01930826.2019.1616971.
- Muslihat, D. (2018). Agile methodology: An overview. Zenkit blog. <https://zenkit.com/en/blog/agile-methodology-anoverview/> B.K Birla Centre for Education, Library. Retrieved from <http://www.iiserpune.ac.in/~library/life2017/program/17/3-sagender-singh-parmar.pdf> on 19/07/2018.
- Ola, Chris O. (2010). The deployment of Virtua Integrated library system software for managing library resources in University of Ibadan, *KDL News*, A Bi-annual Publications of Kenneth Dike Library, University of Ibadan, Nigeria, 14- 15.
- Omeluzor, S. U., Adara O., Ezinwayi, M., Bamidele, M.I., &Umahi, F.O. (2012). Implementation of Koha Integrated Library Management software (ILMS). The Babcock University Experience. *Canadian Social Sciences*, 8(4): 211. Online Library Learning Center Glossary (2006). *Concept of*

Academic Library. Retrieved from <http://www.usg.edu/galileo/skills/olcglossary.html>.

Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, and Dave Thomas, “Manifesto Satpathy S and Maharana R. (2011). ICT skills of LIS professionals in engineering institutions, *Library Philosophy and Practice*, Vol 3: 124.

Sutherland, J., & Schwaber, K. (2017). The definitive guide to Scrum: The rules of the game. Retrieved from: <https://www.scrumguides.org/docs/scrumguide/v2017/2017-ScrumGuide-US.pdf>.

Stoddard, M.M, Gillis B & Cohn, P. (2019). Agile Project Management in Libraries: Creating Collaborative, Resilient, Responsive Organizations. *In Journal of Library Administration*. <https://doi.org/10.1080/01930826.2019.1616971>.

Tyagi, N.(2020). What is Agile Software Development? Agile SDLC <https://www.analyticssteps.com/blogs/what-agile-software-development>

Ukachi, N.B., Nwachukwu V.N. and Onuoha, U.D (2014). Library Automation and use of open source software to maximize library effectiveness. *Information and knowledge management*, 3 (4): 74-80.