THE INTERNATIONAL JOURNAL OF
SCIENCE & TECHNOLEDGE

Multivariate Analysis of Performance of Hybrid Result System Model for Nigerian Universities

Chukwuemeka Etus
Lecturer, Department of Information Technology,
Federal University of Technology Owerri, Nigeria
Gerald C. Eheduru
Professor Emeritus, Department of Information Technology,
Federal University of Technology Owerri, Nigeria
Udoka F. Eze
Professor, Department of Information Technology,
Federal University of Technology Owerri, Nigeria
Charles O. Ikerionwu
Lecturer, Department of Information Technology,
Federal University of Technology Owerri, Nigeria
Obi C. Nwokonkwo
Senior Lecturer, Department of Information Technology,
Federal University of Technology Owerri, Nigeria

Abstract:
Result computation and management in tertiary institutions is one of the key components to ascertain the level of academic quality a graduand possesses. Research indicates that result computation through the available single-platform system models are about 40% deficient (for the offline) and about 50% deficient (for the online), and inadequate to drive the NUC's goal of attaining higher global ranking and competitiveness. To solve this inadequacy, this work developed a hybrid result processing and management system (HRPMS) for Nigerian Universities. The study adopted Agile-Scrum methodology, which provided the platform for quick response time to changes in user requirements and enhanced the pace of development of the software product. In the design and development process, the HRPMS input-output modules and computing components were developed with the following tools: UML designs, Excel Binary VBA, Java 8.0, PHP Designer 8.0 and MySQL5.4 programs, which optimized the standalone-internet hybrid functionalities of the socket-subsystem. Using the Agile fourth-quadrant tests in the beta test stage, the HRPMS performance was validated. The hypotheses test and the probability density function (PDF) prediction value (1.043458X10-8), shows that the non-functional requirement variables actually and positively drives HRPMS performance to becoming improved over the existing single-platform system model. Finally, the HRPMS promises to provide more efficient, more secured and hard-to-crash system for result processing and management in Nigerian Universities.

Keywords: *Single-platform, HRPMS, Agile-Scrum, Socket-subsystem, fourth-quadrant, validation, Cross-platform, Hard-to-crash*

1. Introduction

In Universities, processing and management of students' results involve many complex stages (Okebukola, 2010). As more information are made available in a variety of formats and sources, processing and management of data becomes more and more complex, which can be handled using Information System approach (Oliveira, Thomas & Espadanal, 2014). Despite the proliferation of vendor result processing and management systems, the automation level of the existing single-platform system models prevalent in Nigerian Universities are unsatisfactory. The existing single-platform systems are the completely offline (standalone) system and the completely online (client-server) system, which can be web-based or application-based (Obiniyi & Ezugwu, 2010; Youh, 2010; Eludire, 2011; Ukem & Ofoegbu, 2012; Obasi, Nwachukwu & Ugwu, 2013; Akinmosin, 2014; Anyiam, 2017). Therefore, this work has adopted the hybrid model, to optimize the strengths of the existing single-platform model. In this study "hybrid" refers to an offline-online platform created by combining and optimizing the standalone and the client-server platforms. Each one of these platforms complements the strengths and weaknesses of the other. Hence, the proposed system is a hybrid of the standalone and the client-server, which is a cross-platform architecture. The existing systems used the code-and-fix, waterfall, spiral, staged delivery, evolutionary prototyping, design-to-schedule, and crowdsourcing software development processes. As good as these processes are, they have not always produced reliable software (Ferro, Loukis, Charalabidis & Osella, 2013). So, this study shifted focus to the Agile-Scrum process to develop the HRPMS. This attempts to address the concern raised in the Standish Group Chaos Report 2015 as discussed in Hastie and Wojewoda (2015), on the need to improve the low success rate (29%) and alarming failure rate (71%) of single-platform Information Systems (IS), which has contributed to the poor institutional acceptance and usage of some IS (Graham, Woodfield & Harrison, 2013).

1.1. Problem Statement

In Nigerian Universities, only single-platform (standalone or client-server) result processing and management systems are available today. The single-platform system has proved to be inadequate to handle the results of the growing students' population in Nigerian Universities, thus contributing to their poor ranking, poor global competitiveness and failure to meet the NUC's objective of achieving world-class Universities in Nigeria. This is due to its inadequacy to handle all-round system and data security and confidentiality, in relation to the need for its global accessibility and scalability to institutional formats, policy-standards and regulations. Hence, the need for a hybrid model to improve the weaknesses of the single-platform model, computerizing and automating the results processes in line with global best practices. Other challenges to be addressed by the HRPMS includes: avoidable human errors, unnecessary delays in producing transcripts and litigations arising there from, abandonment of the legacy results due to the difficulty in recomputing them to fit into the existing systems database, incomplete database archive of academic records over the years. Others include incessant system failures (partially or fully), high rate of abandonment of the failing systems, and loss of human and financial resources in developing and deploying the failed systems.

1.2. Objectives

The broad objective of this work is to develop a hybrid results processing and management system (HRPMS) model to improve results computerization and its process automation in line with institutional policy-standards and authority-lines in Nigerian Universities. The specific design objectives of this work are:

- To do survey analysis of the existing single-platform model based on the performance variables to identify the weaknesses necessitating the HRPMS.
- To model the HRPMS input-output design using the Unified Modelling Language.
- To develop the HRPMS input-output modules using Excel VBA Binary, Java 8.0, PHPDesigner 8.0, and MySQL5.4 Database Management System.
- To test and validate the performance of developed HRPMS system model statistically.

1.3. Hypotheses

The following null hypotheses were tested in this work:

- H_{01}: The non-functional variables have no significant effect on system performance.
- H_{02}: There is no significant difference between the existing and proposed system performance.

2. Literature Review

2.1. Conceptual Framework

The proposed HRPMS framework conceptualizes and organizes the system development, and captures the high points of the system solution. It presents how the proposed system has been modeled to isolate and selectively implement the result processes, some on the standalone subsystem modules and some on the network subsystem modules, to optimize their strengths and weaknesses. Hence, it addresses the research gap.

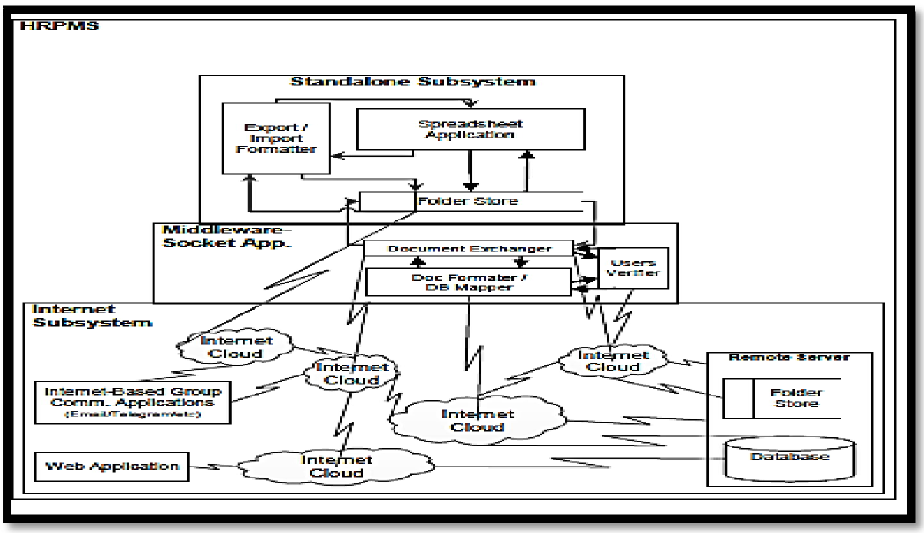

Figure 1: The Hrpms Model Conceptual Framework (Hybrid Structure)

As shown in Figure 1, the HRPMS model conceptual framework consists of the standalone and internet subsystems, interfaced using the middleware-socket application. The standalone subsystem comprises the spreadsheet, export/import formatter, and folder store. Its functions include enlisting students and courses, input/output formatting, results computation and reporting, reference-courses tracking, and one-click transcripts generation. The implementation represents that of the more critical and more scalable functions. The Network subsystem comprises the internet, internet-based group communication and web applications, remote server housing database and secured group communication folders, data and files. It is used to transmit computed results for vetting, approval, and upload to database and to the approved results folder for results checking and transcripts processing. The functions are less critical and less scalable. The middleware-socket application comprises the users-verifier, document-exchanger, document-formatter-database-mapper modules, and PINs generator for results checking. The functions authenticate users and their access privileges (for moving result files through the university management for vetting, approval, and retention), formatting and mapping of result files and results data for upload to the database, and finally retaining the approved result files in approved results folder. Implementing the HRPMS model would ensure distributed processing, efficient and secured group communication for results transmission. It will also ensure redundancy of approved results.

2.2. Theoretical Framework

This section presents the theories for constructing, deploying and validating the proposed system. The framework organizes the system components, its developments, integration and empirical analysis. It captured the realities of the solution principles, ideas and context for this work. The frameworks utilized in this work relates to the legacy system, spreadsheet standardization, the platform optimization, and the dual-validation using the system functional and non-functional requirements. Based on these, the research was realized.

2.2.1. The Legacy-Modern System Hybrid

Legacy systems are systems that have outlived their original user requirements but have remained in operation long enough to be substantially modified until the system no longer resembles that which was first developed. However, legacy-modern system hybrid concept (Desouza, 2003), upholds that the maintenance process continues because the system functions correctly, even though a large percentage of the code is obsolete or the remainder frequently works in ways that are not fully understood by those maintaining it, or that the system needs some newer features and functions. According to Robertson (1997), IS organizations are struggling to respond to demands for new features on existing systems whilst simultaneously being expected to incorporate new technologies. Sometimes, this leads to the challenge of trying to enhance legacy systems without being fully sure how they work, and trying to use newer methodologies to improve systems originally developed following the traditional 'waterfall' model, making enhancements to legacy systems tiresome.

The best way to retain the functionalities of a legacy system and also add some cutting-edge functionalities is by hybridizing legacy and trending systems (Avison, Lau, Myers & Nielsen, 1999). With this, some system functionalities are further modified to accommodate businesses, structural or technical changes such as new facilities, policies, trends, and so on; which its absence can often lead to loss and/or wastage of resources, opportunities, time and efforts. In this work, the researcher reworked a legacy system and hybridize with a modern system to realize a state-of-the-art mainstream system (HRPMS) with desirable old and new abilities and functional features. This was done using modern development methodology in an attempt to meeting changing user needs, meeting changing policies and mandates by the Nigerian Universities Commission (NUC), and meeting the challenge of improving online presence and visibility, impact factor and global ranking of Nigerian Universities.

2.2.2. The Spreadsheet Standardization

The concepts reviewed here are relevant to the data processing and storage strategies adopted in this work to develop the data upload and download modules to and from the database. The Microsoft Excel is a spreadsheet platform used to input and output tabular data (Spreadsheet Analytics, 2010). A major drawback of the excel sheet file is that it is a domain specific tool and must be standardized to interface well with other systems (Grossman, Mehrotra & Ozluk, 2007; Grossman & Ozluk, 2011). Standardization is the only means of providing a more stable and reliable spreadsheet software product (Powell & Baker, 2009). To standardize a draft of the excel interface, the platform design should provide a strategy for integration (Grossman, 2006). This facility creates an opportunity for effective exchange and flow of information between interfaces – offline and online.

Standardization ensures the deliverables of data harvest by using standard codes to select the data on the excel sheet. This is known as data analysis concept (Agard & Kusiak, 2004). It involves doing invariant based programming for per record processing to get essential data per record (Raffensperger, 2001). Top-down programming by contract is also a standardization concept used by hybrid-platform developers to prepare offline data for uploads to online applications, and vice versa. Its protocols are (Sun Microsystems, 2004): (i) The Excel interface must be a single excel (for example, a single excel file sheet must contain the result for a class of students for a particular semester and year); (ii) The fields available must be consistent throughout an excel sheet; (iii) There are to be no free columns (a free column is a redundant column that lacks semantic presence, serves no purpose, useless, and must be eliminated); (iv) The fields must also be contiguous from row to row (once a data field begins, it must run to the end with no unused row in-between the ends).

These suggest that the users of the core part of the system should understand the structure of a standardized spreadsheet and the data layer designs accommodating the type of data being designed for. This contract must be understood so as to allow for better program performance. Hence, standardization is critical to managing electronic records effectively (Asproth, 2012). The key to an offline-online hybrid system lay not in using the simple functions open to simple spreadsheets, but in a critical analysis of the sheet (Raffensperger, 2001; Spee, Jarzabkowski & Smets, 2016). It is essential to consider some facts about a typical spreadsheet result namely: (i) There can be a variable number of results per semester per level; (ii) Given that we know the number of courses we can determine the number of cells to process per student; (iii) There exists a cell with exactly eleven digits, immediately after this cell comes a name string followed by a known number of cells; (iv) It is possible to maintain a parallel array of global data for the results and units; (v) It is possible to do a one-time processing based on the business logic design adopted. The HRPMS offline sub-system (the result processor-formatter sub-system) was developed using the spreadsheet standardization concept and deployed as standalone system. Also, the online sub-system (the converter-database sub-system) was developed by spreadsheet analysis concept implemented using package by feature processing and deployed over the Internet. The combined offline-online (standalone-Internet) system is founded on the hybrid concept.

2.2.3. The Platform Optimization

This concept is relevant to maximizing the inherent strengths of separate single platforms combined while minimizing their weaknesses, to effectively deploy the proposed system. There are four (4) major single platforms for deploying software applications world over. They include the standalone, private-cloud, community-cloud, and public-cloud platforms; and combining any two, or at most all of them, result to hybrid platforms (Shin, 2016; Labra et al., 2016). As single-platform systems, the standalone, the private-cloud-based, the community-cloud-based, and the public-cloud-based systems all have their inherent strengths (sufficiencies), weaknesses (deficiencies), and some other demands (user and system requirements) (Shin, 2016; Labra et al., 2016).

Based on empirical review of available literatures (Obiniyi & Ezugwu, 2010; Youh, 2010; Eludire, 2011; Ukem & Ofoegbu, 2012; Obasi, Nwachukwu & Ugwu, 2013; Akinmosin, 2014; Anyiam, 2017), the major strengths of the standalone result processing and management system include good security and confidentiality (privacy), while its major weaknesses include non-global presence and access due to no-network. The private-cloud-based and community-cloud-based systems could ensure fairly good security and confidentiality, but provides location-limited presence and access due to limited coverage. Also, the strength of the public-cloud-based systems includes global presence and access, while its weaknesses include relative security and confidentiality due to varying degrees of security threat and challenges. These inherent strengths and weaknesses are therefore critical to the success and failure of single platform information systems (Ward & Peppered, 2016).

To retain the strengths of single-platform systems while reducing their weaknesses, there is need to combine and optimize them (Popovic, Hackney, Coelho & Jaklic, 2012). These can be done through hybridizing the single-platform systems, through which these strengths and weaknesses can be selectively addressed (Hemmatian, Fereidoon, Sadollah, & Bahreininejad, 2013; Zhang, Wang, Dong, Phillip, Ji, & Yang, 2015). Therefore, to deploy the proposed system, the researcher intends to hybridize three (3) single-platforms namely - the standalone platform for offline deployment, with the private-cloud and public-cloud platforms, for online deployment. This hybrid system design and development deployable over a hybrid platform is a cross-platform system. The HRPMS has the task of retaining the strengths and sufficiencies while eliminating or minimizing the weaknesses and deficiencies of the single-platforms systems (with other system performance and acceptance criteria), to take care of the research problems and fill or close the research gap. This system solution could effectively improve the computerization of result processing and management in Nigerian Universities as developed and deployed in accordance with its performance requirements (non-functional specifications or criteria).

2.2.4. The Hybrid – Validation

This concept review is relevant to identifying and establishing the functional and non-functional system requirements for validating the HRPMS development. The functional requirements are the operations or services of the Information System (IS) while the non-functional requirements are the qualities of the operations or services (Dabbagh & Lee, 2014; Bahill & Madni, 2017). While the functional requirements are system oriented only, the non-functional requirements are system, system-environment and user oriented (Eckhardt, Vogelsang, & Fernández, 2016). Whereas system functional requirements describe the software functions or its capabilities taken care of during the software development, the non-functional requirements act as constraints to the software solution in terms of quality, and are therefore used to determine the performance, maintainability, safety, reliability and usability of the system software (IEEE Computer Society, 1998; Abran, Moore, Bourque, Dupuis, & Tripp, 2004; Bhat, Gupta, & Murthy, 2006; Boegh, 2008). These point to the success criteria for total validation of any IS or IT project.

Every project development has its own set of success criteria in which its absence leads to total project failure, and software development is not an exception to this. In other words, successful software development project must intimate its success drivers. According to Eklund and Bosch (2013), software will be successful if it delivers the desired performance improvement, and also delivers cost that provides value for the organization. It is sad news that software and IT projects succeed at a low rate and also fail at an alarming rate around the world (Graham, Woodfield & Harrison, 2013; Fero, Loukis, Charalabidis and Osella, 2013; Oliveira, Thomas and Espadanal, 2014). Standish group Chaos report (2015), as cited in Hastie and Wojewoda (2015), indicated that there is still much work to be done towards achieving successful outcomes from software development projects. According to the report, only 29% of all projects were fully successful, 52% of all projects were cancelled, with a failure rate of 19%; giving a total of 71% of unsuccessful software and IT systems. This report supports the problem statement and research gap in this work.

2.3. Software Development Processes

The software development theory can better be understood by looking at its processes. Software development is concerned with producing software products. A software development process is concerned primarily with the production aspects of software development before the technical aspect, such as software tools. Specific software development tools are useful and often necessary, but the true art in software development is applying the correct process and then using tools to support the method. Without a method, tools are worthless (Karagiannis, 2015).

Following the SDLC as a framework of various software development methodologies, various software development processes were considered based on their various advantages one over another and their peculiarity in solving various software development needs. They include Code-and-Fix Model, Waterfall Model Process, Spiral Model, Staged Delivery Model, Evolutionary Prototyping Model, Design-to-Schedule Model, Crowdsourcing Process **(**Boehm & Turner, 2005; Khan & Beg, 2013). Software development methods and processes are so many that developers' battle with choosing from them for different projects. Practically, most of them are the same in their processes, except for variations to improve the deficiencies in them. To assist in the choice of software development methods and processes, Khan and Beg (2013) rated the models alongside project management techniques, such as risk management, quality and cost control, success predictability, progress visibility and customer involvement. The degree of influence of the management techniques was rated from 1 (least influencing factor) to 5 (most signifying factor), as in Table 1.

Software Development Model	Risk Mgt	Quality/ Cost Control	Predict-ability	Progress Visibility	Customer Involvement
Code-and-fix	1	1	1	3	2
Waterfall	2	4	3	1	2
Spiral	5	5	3	3	3
Evolutionary Prototyping	3	3	2	5	5
Staged delivery	3	5	3	3	4
Design-to-schedule	4	3	5	3	2

Table 1: Software Development Model Matrix (Rated 1-5 For Each Category)
(Khan & Beg, 2013)

From Table 1, code-and-fix showed as a weak project management application technique with poor risk management, quality/cost control, success predictability, with improved project progress visibility, and customer involvement. Evolution prototyping also revealed different levels of project management influence on them. In all, the choice of software development model is guided by the task at hand, risk management, quality/cost control, and predictability of the project, visibility of progress, and customer involvement and feedback. In all, the software development model to be adopted in this work is the Staged Delivery which has medium to high range ratings of the signifying factors as in Table 1. A new version of the Staged Delivery development model used is the Agile Model. The Standish group report of 2015 (Hastie & Wojewoda, 2015), had it that using agile approach results in more successful projects and less outright failures, hence its adoption in this work.

2.4. Agile Software Development Process

This review gives an overview of the Agile development method. Rapidly changing environments characterized by evolving requirements and tight schedules require software developers to take an agile approach, instead of the traditional software development approaches (Theunissen, Boake, & Kourie, 2005). The Agile methods include practices such as short iterations, frequent releases, simple and emerging design, peer review, and on-site customer participation (see Figure 2). Also, Agile processes are characterized by iterative development, continuous code integration, and the ability to handle changing business requirements (Boehm & Turner, 2005; Aveling, 2004). Implementations of agile method include the following approaches: Rational Unified Process, Extreme Programming (XP), Scrum, Crystal Clear, Adaptive Software Development, Feature-Driven Development, Test-driven Development, Lean, Kanban, and Dynamic Systems Development Method (DSDM).

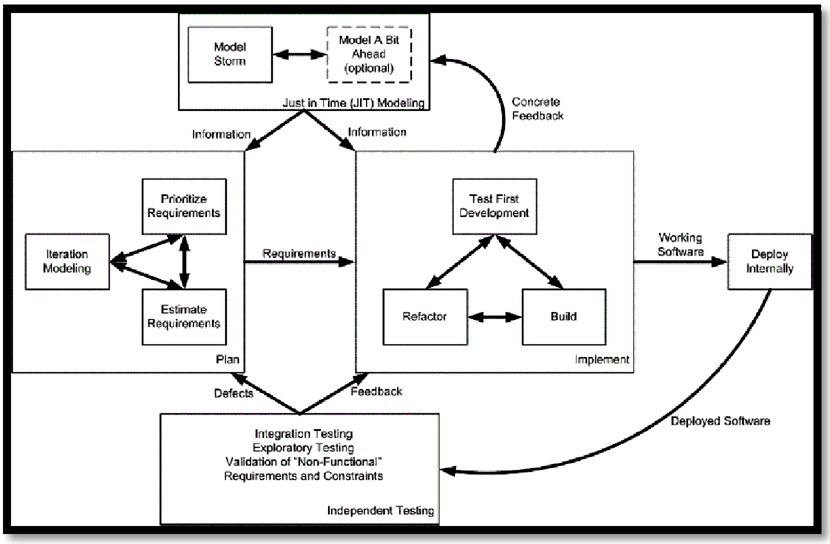

Figure 2: Agile Development Process (Theunissen, Boake, & Kourie, 2005)

The Agile process has many techniques. The Agile Manifesto collectively refers to these techniques as the agile methodologies (Fowler & Highsmith, 2014). The Manifesto holds that, while we value the items on the right, the items on the left is valued more. The values highlighted (Agile Alliance, 2001) are: (i) Individual interactions over processes and tools; (ii) Working software over comprehensive documentation; and (iii) Customer collaboration over contract negotiation; (iv) Responding to change over following a plan. The Agile Scrum method was adopted rather than Extreme Programing (XP), because it manages the entire software project rather than just the development practices.

2.5. The Scrum Approach

According to Leau, Loo, Tham and Tan (2012), in today's software development environment, requirements often change during the product development lifecycle to meet shifting business demands, creating endless headaches for development teams. The Scrum development process model came to address this concern of changing requirements. Scrum is a process for incrementally building software in complex environments (Kroeneke & Hattach, 2000). It provides empirical controls that allow software development to occur as close to the edge of chaos as the developing organization can tolerate. Scrum is a software development process for small teams. According to Reeder (2015), small teams that work independently are more effective. Scrum is basically an incremental, sprinted (time-boxed) process with an added twist: the frequent meetings to review and address the risk elements.

Scrum is appropriate for software developments where the requirements cannot be defined up front, and chaotic conditions are anticipated throughout the software development life cycle. Consequently, the Scrum process results to: products that become series of manageable chunks, progress made even when requirements are not stable, everything being visible to everyone, team communication improvement, team sharing successes along the way to the end, customers seeing on-time delivery of software increments, customers obtaining frequent feedback on how the product actually works, customer (user) relationship that builds trust and grows knowledge, and a culture created where everyone expects the project to succeed. Hence, the Scrum approach has been defined as a flexible, holistic product development strategy, proposed to result in fast, flexible and holistic product development (Reeder, 2014).

In Scrum, cross-functional teams develop products or projects in an iterative, incremental manner. Development is structured in time-boxed cycles of one (1) to four (4) weeks work (called Sprints), which takes place one after the other without pause for a few months. The Sprints (time boxes), end on specific dates whether the work was completed or not, and are never extended. Usually Scrum Teams choose one Sprint length and use it until they improve and can use a shorter cycle. At the beginning of each Sprint, a cross-functional Team selects items (requirements) from a prioritized list. The Team agrees on a collective target of what they believe they can deliver by the end of the Sprint, something tangible is "done" (Deemer, Benefield, Larman, & Vodde, 2012).

During the Sprint, no new items may be added; Scrum embraces change in the next Sprint, but the current Sprint is meant to focus on a small, clear, relatively stable goal. Every day the Team gathers briefly to inspect its progress, and adjust the next steps needed to complete the work remaining. At the end of the Sprint, the team demonstrates what it has built and obtain feedback that can be incorporated in the next Sprint. Scrum emphasizes working product at the end of the Sprint that is really "done". In the case of software, this means a system that is integrated, fully tested, end-user documented, and potentially shippable (Deemer, Benefield, Larman, & Vodde, 2012). The key roles, artifacts, and events are summarized in Figure 3.

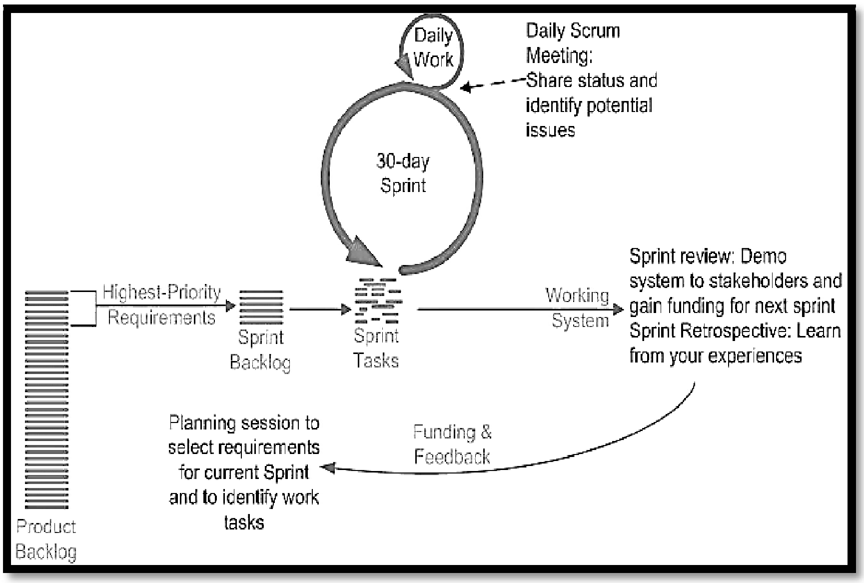

Figure 3: Overview of the Scrum Approach (Deemer Et. Al., 2012)

Every Scrum team has three roles: Product Owner, Scrum Master, and Team Development. Individuals in these roles work together to bring a product from an idea to life (Deemer et. al., 2012; Reeder, 2014). Scrum teams are self-organized and cross-functional. Self-organized because rather than directed, they choose how best to accomplish their work. Cross-functional because they have all competencies needed to accomplish the work without depending on others not part of the team. The Scrum team is designed to optimize flexibility, creativity, and productivity. Finally, Scrum includes five activities that make for transparency and adaptation: Product Backlog refinement, Sprint planning, Daily Scrum, Sprint review and Sprint retrospect (Schwaber & Sutherland, 2015; Schwaber & Sutherland, 2014). Before the sprint comes the Product, Backlog containing the Vision Statement, the product roadmap, and the deliverable features. These provide a concise description of the goals of the project for focus, an initial visual timeline of major product features to be delivered, and the requirements that make them up. All these stories make up the Product Backlog. Scrum do not wait until the Product Backlog is 100% prepared with all the details to start the Sprints, but starts the Sprints as soon as the Product Backlog is mature enough and has enough stories defined. Product Backlog would be kept updated during the project (Rad & Turley, 2013).

According to Philips (2014), Scrum has the benefits of divide and conquer, inspect and adapt, and transparency. By divide and conquer, Scrum divides complex work into simple pieces, large organizations into small teams and far-reaching projects into a series of short time horizons called sprints. Simple pieces make it easier to map out what needs to be done. With a clear roadmap, the team starts work immediately, knowing what items needs to be worked on together and when the job has been completed. Sprinted projects and small teams make it easier to maintain focus because less time is spent coordinating and communicating details. It is easier to plan for short periods of time involving less. By Inspect and Adapt, the team gather feedback more quickly from completed and functioning product in the hands of users than it otherwise would have. Feedbacks help the team improve the product based not only on what they have learned during development, but also from people interacting with the product. By Transparency**,** Scrum uses shared tools called Artifacts, and regular meetings called Ceremonies to track and visualize activities over the length of the sprint and also facilitate team communication. This way, it is easy for everyone to know how things are moving along - what needs to be done, who is doing it, and how it is being accomplished. Stakeholders and management are able to make more informed strategic decisions when they have an honest and clear idea of how a project is progressing.

2.6. Empirical Review

This section specifically reviews related students' result processing and management systems in some Nigerian Universities, with available and properly documented students' result systems. They include Ahmadu Bello University (ABU) Zaria, Gombe State University (GSU), Joseph Ayo Babalola University (JABU), University of Calabar (UNICAL), University of Portharcourt (UNIPORT), Nasarawa State University (NSU) and Federal University of Technology Owerri (FUTO). The reviews showed the existing systems, their development drivers, their design, their implementations, and their assessments that gave rise to the research gap identified in this work.

2.6.1. The ABU Zaria Result Processing and Management System

According to Obiniyi and Ezugwu (2010), their work at Ahmadu Bello University (ABU) Zaria identifies the causes associated with delays in processing and releasing result in Universities, and proposed enhanced computer program for result computation integrated with a database for storage of processed result. This simplified the university grading system and overcame the short-comings of the existing packages. The system took interdepartmental collaboration and alliances into consideration, over a network that speeds up collection of processed result from designated departments through an improved centralized database system. This expedites processing of result and transcripts at various levels of management and access to student result online.

The technological approach for the implementation of the system was based on open source solutions – the crowd sourcing development process. Apache was used as the Web server extended with PHP for server-side processing. In recognition of the confidentiality of data contained in the system, communication networks were protected with open-ssl library for data encryption and role-based authentication. This system increased the service delivery efficiency and benefited both the administration and students. However, this system suffered from online security threats on the editing and administrative rights. This hampered its performance and institutional usage.

2.6.2. The GSU Students Result Processing System

In the work done by Youh (2010), today's university environment has an increasing need for distributed database systems as the desire for easy, reliable, scalable, and accessible information is steadily increasing. The inadequacies of centralized database systems in handling students' result necessitated the use of distributed database systems in that work. According to the author, the system was a relational database designed in a way that each academic department in the university had its own database including the Central Record Processing Unit (CRPU), Exams and Record Unit, Student Affair Division, Dean's Offices, and Senate. The master database was hosted at CRPU. Also, in the system development, Microsoft Visual Basic 6.0 and Structured Query Language (SQL) were used to design a prototype of a client server distributed database system for processing student exam records. But this system had drawbacks in limited non-global presence for access by users, since the system was not accessible over the internet. This also hampered its institutional usage.

2.6.3. The JABU Students Academic Record Management System

A number of problems bedevil the student academic record management, and they include improper course registration, late release of students' result, inaccuracy due to manual and tedious calculation and retrieval difficulties and inefficiency. In most cases the data generated by academic institutions are usually created in non-delineated files for use by different departments and units within the institution with the same data appearing on several of these files. This meant that a simple change of address would have to be processed in two and probably three or four places, depending on the number of other files on which these data appears. Hence, according to Eludire (2011), the development of database concept is the answer to the above-mentioned problems, where the amount of redundant data is reduced, with the possibility of inaccurate data contained on a file because they were never updated.

Therefore, Eludire discussed the design and implementation of a student registration and course management database application with Microsoft Access. The work also discussed the issues of selecting appropriate database model, interface design, system deployment and maintenance. A projection of record growth in relation to student population and system requirement was carried out in the study. Finally, the work discussed the applicability of the system in academic institutions. This system was developed using the evolutionary prototyping model development process. Consequently, the system could only be deployed as standalone since the Microsoft Access used standalone database, and the deployment could not achieve global presence. This hampered its institutional deployment and usage.

2.6.4. The UNICAL Result Processing and Management System

In the work done by Ukem and Ofoegbu (2012), a computer software application was developed to facilitate the automated processing of the result in the University of Calabar, Nigeria. The software was developed following the code-and-fix development process, with Java programming language and the database was designed by employing MYSQL Relational Database Management System. The developed software performed well and produced expected result on completion. With it, it was possible to compute Grade Point Average and Cumulative Grade Point Average for each student based on examination scores entered. This system was only a standalone system and hence, deficient in global accessibility, especially by students to view their result. It also had the tendency to be vulnerable to hacking should it be hosted online, due to the editable administrative privileges on it, amongst other security threats. This hampered its institutional usage.

2.6.5. The UNIPORT Result Processing and Management System

The UNIPORT system also presented a single platform used to manage and process data for all categories of students in a seamless and interactive manner. The design technology adopted for the implementation was a client/server technology, with MYSQL as the server technology and Visual Basic.NET as the client technology. Internet Information Server (IIS) was used as the Web server. The software development methodology adopted was the incremental model in conjunction with prototyping technique (evolutionary prototyping development process) (Obasi, Nwachukwu and Ugwu, 2013). An empirical evaluation of the system showed that the system expedites processing of students' result and generation of other related academic information and increased efficient service delivery with added advantage of academic records management. Nevertheless, the system was web-based and run on the public-cloud, and thus bedeviled by the hacking and unauthorized access vulnerability. These have seriously limited the actual usability of the system to 100 level students' result computation, beyond which the system suffers heavy attacks.

2.6.6. The NSU Result Processing and Management System

This research done in Nasarawa State University, Keffi Nigeria, was focused on creating an automated student result management system using Oracle database, forms and reports. Akinmosin (2014), in the work, showcased a computerized examination result management system for tertiary student's examination records. According to the researcher, the manual method of students' academic result processing was found to be tedious, especially when carried out for a large number of students, and this also makes the entire process time-consuming and error prone. The system designed was meant to register students as soon as they have paid their departmental registration and only then will they be able to view their result. The system presented a private-network-based single platform to be used to manage the processing of all examination records within the institution. The research implemented an application for student result storage, and users can access this software from anywhere as long as they are working within the network coverage of the application server.

This application was designed using Oracle database for keeping the records (data), Oracle forms for creating the user interface, and Oracle reports for presenting information to the user, having been developed through the waterfall process. The system enabled users with the right permission to insert new user details, student's examination records, as well as update these records whenever necessary. The review showed that the system expedites the processing of students' examination result and its reporting. Nevertheless, users of the system need certain level of Oracle related skills, and they need to be present in the institution to connect to the network before using the system. Poor user-friendliness and physical presence, amongst other challenges, are the constraints to the actual usage of this system.

2.6.7.The FUTO Result Processing and Management System

This work revealed the result processing and checking challenges presently faced by the Federal University of Technology Owerri. The development used the Structured System Analysis Development Methodology (SSADM) to design and implement an enhanced web-based students' result processing and management system with result checking ability, in the bid to remedy the freezing challenge of the existing system. This development method provided unambiguous identification of users by means of their username and password. It provided a platform for easy course registration by students, result collation, grading and publishing by lecturers, result checking by students, and management decisions by school authorities and the exams and records unit. The work produced a software with SQL Server 2012 DBMS to aid result collation, processing, publishing and checking (Anyiam, 2017). However, system development, management and maintenance were completely outsourced; yet the issues of security threats to the database and data redundancy were poorly addressed. These made the system vulnerable to cyber-attacks capable of grounding it any time.

2.6.8. The Research Gap

The empirical review of literatures in this work shows the research need to explore and use the under-explored hybrid system model for results processing and management. This has become necessary since efforts at improving the single-platform results model can neither eliminate its inherent weaknesses nor effectively address its attendant effects. These challenges include: (a) delayed processing of results offline; (b) difficulty computing legacy and borrowed results, and doing complex result corrections online; (c) poor scalability and compliance to formats, institution-specific policy-standards and regulations; (d) security and confidentiality threats that include privilege escalation, SOL injection, phishing, DoS session hijack, wireless cracking, back-door & blended attacks; and (e) little or no database of students' results for past years. The immediate and remote effects of the weaknesses on the existing system include: (a) poor institutional usage; (b) incessant system failure (partial or full); (c) high rate of abandonment of the failed systems; and (d) loss of human and financial resources. Thus, the single-platform result system model proves deficient and inadequate, necessitating another model that will optimize the strengths of the offline and online systems.

Research (cite) indicates that the gap is part of the metrics used in University ranking, as such, Nigerian Universities have been ranked poorly globally. To bridge the gap between using the single and hybrid platform in results processing and management, the HRPMS was developed and validated following the Agile-Scrum development and Agile four-quadrant test processes following the functional and non-functional user requirements. The proposed system consists of interoperable cross-platform offline-online system modules deployable over offline (standalone) and online (client-server internetworked) platforms. This helps to address the attendant inadequacy of the single-platform result system model, and optimize its strengths and weaknesses. The validation tests evaluate system operations, user requirements, cost effectiveness, and system performance improvement in alpha and Beta stages, both technically and statistically. The functional and non-functional requirements were selected considering the user and the system specifications for the HRPMS.

3. Methodologies

3.1. Preamble

This chapter presents the methodology used in this research for the HRPMS development and validation. It shows the technical methods or techniques, procedures and tools for the HRPMS design, coding and deployment. It also shows the technical and statistical methods, procedures, and instruments for evaluating and validating the HRPMS development. These were put together in the HRPMS thesis structure. The HRPMS thesis structure is a research design framework that guided its development. It ensured that the work done is economical, accurate and consistent with the study objectives, and it demanded answers to questions about relationships and generalization of results in line with the standards (Madukosiri & Bawo, 2012; Petroshius, 2015). These showed that a thesis structure (also called research design) models a research as a systematic study outline or a researcher's compilation method detailing how the study will arrive at its logical conclusions (Creswell, 2013; Maxwell, 2012; Yin, 2013). Hence, to develop the HRPMS thesis structure, the researcher studied the existing systems (pilot study) to verify processes and devices, monitor facilities and activities involved, and make relevant decisions. The pilot study and the literature reviews helped to refine the design to arrive at the research framework. Figure 4 shows the HRPMS thesis structure comprising topic selection, literature reviews, problem statement, identifying the research gap, identifying the functional (operational) and non-functional (performance) requirements, and creating the research model. Others are selecting the materials and methods, analyzing the existing single-platform systems, HRPMS development and validation through technical and statistical evaluations.

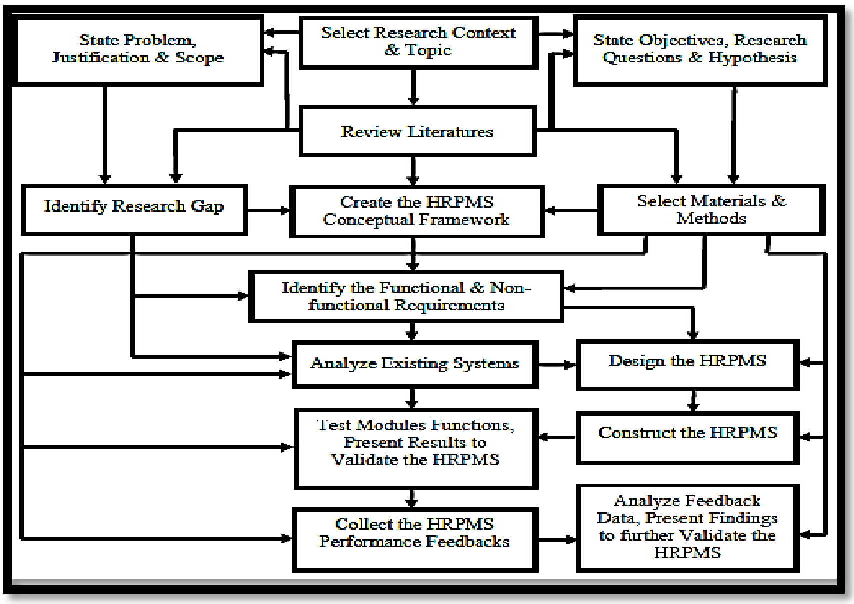

Figure 4: The HRPMS Thesis Structure

3.2. System Analysis and Development Method

3.2.1. Existing System Analysis Technique

The pilot survey method (using the qualitative technique), was deployed in this work to systematically analyze the existing single-platform system to determine their weaknesses. The qualitative technique is easy to use and produces precise results (Mills, Teplitsky, Arroyo, Charmantier, Becker, Birkhead & Bushuev, 2015; DaSilva, Neto, O'Leary, deAlmeida & deLemosMeira, 2015). Interview and observation tools were to gather the pilot data, which were tabulated to show the results and inferences. The High-Level Model (HLM) diagram was used to present the existing systems' structural overview.

3.2.2. HRPMS Analysis and Development Technique

The Agile Scrum Method was followed to develop the HRPMS modules. This is because it accommodates structured and object-oriented analysis and design methodologies. It focuses on holistic management of the software development project rather than just the activities or practices (Deemer, Benefield, Larman, & Vodde, 2012; Herzberg & Sisombat, 2013; Rad & Turley, 2013; Reeder, 2014; Schwaber & Sutherland, 2015). The Agile Scrum method was chosen because it requires less planning, divides and handles tasks in small iterative increments, and reduces errors (see Figure 5). Hence, it is a quick and efficient way of achieving modularity, increased flexibility, software interfacing with respect to the design, changeable requirements according to user needs with new features easily added (Masila, 2014; Michaels, 2013; Leau, Loo, Tham & Tan, 2012). Customer satisfaction is the highest priority of the Scrum Agile method; hence, it promotes interaction throughout the development process cycle, and customers are directly involved in software evaluation. This make it adaptive in planning, iterative in process and time boxing in duration. The outcome of the subsequent iteration is an enhanced working product. This is repeated till all functionalities are accomplished.

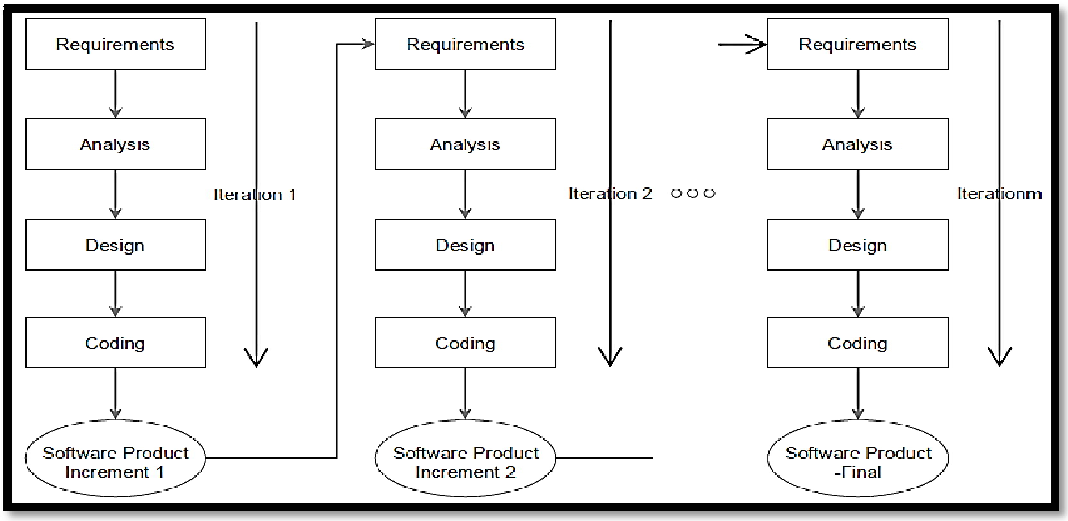

Figure 5: Scrum Agile Software Development Model (Michaels, 2013)

The Scrum Agile procedure covers every system element, from logical to physical modeling, and the implementation of the system processes (Isaias & Issa, 2015; Esiefarienrhe, Adeiza & Ejura, 2015). It yields better software as attested to by the Standish group chaos report in 2015 and 2016, which had it that the agile software approaches resulted in more successful projects and less outright failures. Also, according to SquareSpace in 2013 as cited by Herzberg & Sisombat (2013), more than 70% teams across IT industries are practicing Scrum Agile procedure generally. Even, ccompanies like Microsoft, Adobe, CA, and HP, use Scrum Agile in their software development. In line with the Agile procedure, the Unified Modeling Language (UML) was used to create structural, behavioral, logical and physical models of the HRPMS modules and database, using the High-Level Model, Use case, Flowchart, class, entity-relations, and deployment diagrams (see Figure 6). According to Ahmed (2010), these diagrams were used because of their simplicity in system representation, apt to communicating system logics, efficiency in problem analysis, and ease of documentation.

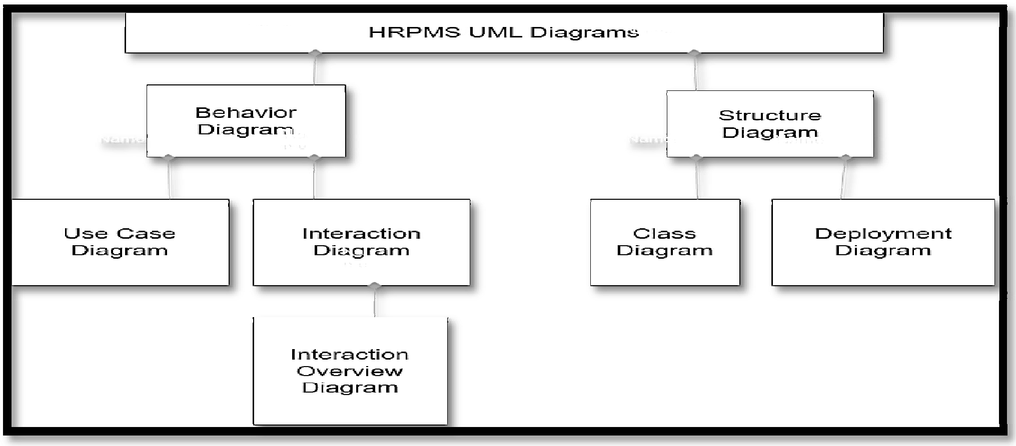

Figure 6: The UML Diagrams Used In HRPMS Design

3.2.3. HRPMS Development and Deployment Tools and Materials

The hardware tools are personal computers and modems, while the software tools include the Excel VBA, JDK8.0, JRE (SE), MS POIs, Browsers, and MYSQL DBMS, and WAMP (local) server and GoDaddy (live) server. Following the top-down technique, the HRPMS modules were developed. Thus, the standalone modules used Excel Visual Basic Application programming, the internetworked modules used Java programming (NetBeans 8.0 Coder, JDKv8.0 development kit, and MS POIs third party library package), and the database module used MySQL5.4. Also, the MySQL connect, the Hypertext transfer protocol (HTTP), and file transfer protocol (FTP) were used to link, manage and transfer data between the computers and the database servers.

3.2.4. HRPMS Functional and Non-Functional Requirements

From the pilot survey and contents review of the user requirements, the nine (9) HRPMS functional (Operational) requirements are: Course Details Capturing, Student Details Capturing, Results Computation, Reports Generation, Reference Tracking, Results Retention Processing, Results Checking, Transcripts Processing, and System Administration. These are described in Table 2. These represent the basic functionalities of the HRPMS.

S/N	Operational Drivers	Descriptions
1.	Course Details Capturing	System captures all core, elective and borrowed courses and units for all students for all sessions
2.	Student Details Capturing	System captures all student's personal and academic clearance details for all student levels.
3.	Results Computation	System auto-computes all student's GPAs and CGPAs from their grade results per semester
4.	Reports Generation	System auto-generates summaries per semester and graduating students' status list per class
5.	Reference Tracking	System auto-tracks all omitted and carryover outstanding courses per student for advisory use
6.	Results Retention Processing	System forwards results to management for routine vetting and final approval and retention
7.	Results Checking	Stakeholders use system to access approved results
8.	Transcripts Processing	Management uses system to generate and transmit academic transcripts to return addresses
9.	System Administration	Administrator manages database, users and PINs to forestall total data loss and ensure good control

Table 2: Description of HRPMS Functional (Operational) Requirements
(Field work)

Also, from the pilot survey and literature review of the system quality requirements (Mangin, Bourgault, Guerrero, & Egea, 2011; Mana & Kanthawongs, 2012), the five (5) nonfunctional (performance) drivers of the HRPMS are Usefulness, Ease of Access and Use, Security and Confidentiality, Processing Speed, and Computation Accuracy; as described in Table 3. These represent the basic capability constraints or qualities of the HRPMS.

S/N	Performance Variables	Descriptions
1	Usefulness	degree to which user productivity is enhanced by the system
2.	Ease of Access and Use	the degree to which accessing and using the system is free of much efforts and skills
3.	Security and Confidentiality	the degree to which the system, its data and user's privacy cannot be compromised
4.	Processing Speed	degree to which system use conserves time and reduces delay
5.	Computation Accuracy	the degree to which system processes and computation outputs can be trusted to be correct and free from errors

Table 3: Description of HRPMS Non-Functional (Performance) Requirements
(Field Work)

3.3. Statistical Analysis Method

3.3.1. Source of Data

The data for this research was collected from nine (9) universities in South-East Nigeria. As shown in Table 4, nine (9) out of the 23 Universities in South-East Nigeria were purposively selected using one-stage cluster and stratified sampling technique to cover all mix of three (3) classifications, namely: generation (first, second, and third generations), specialty (conventional, science/technology-based, agriculture-based, and education-based), and ownership (federal, state, and private Universities).

S/N	University	Owner-Ship	Gener-Ation	Specialty
1	University of Nigeria Nsukka (UNN)	Federal	First	Conventional
2	Nnamdi Azikiwe University Awka (NAU)	Federal	Second	Conventional
3	Federal University of Tech. Owerri (FUTO)	Federal	Second	Science/Tech
4	Michael Okpara Uni. of Agric.Umudike (MOUA)	Federal	Second	Agriculture
5	Federal University Ndufu-Alike (FUNA)	Federal	Third	Conventional
6	Imo State University (IMSU)	State	Second	Conventional
7	Enugu State Uni. of Science Tech (ESUST)	State	Second	Science/Tech
8	Anambra State University Uli (ASU)	State	Third	Conventional
9	Madonna University Okija (MUO)	Private	Third	Conventional

Table 4: Classification-Based Selection of Nine (9) Nigerian Universities
(Field Work)

These were selected for: (i) faster data collection due to proximity to the researcher, and (ii) their educational advancement and (iii) their fair representation of others. Using the Krejcie and Morgan's table for finite but stratified population (Krejcie & Morgan, 1970, as cited in Hill, 1998), the population was reduced from 9,144 (Table 5) to 2009 (Table 3.5). The target respondents were academic and registry staff who produce results.

STAFF	UNN	NAU	FUTO	MOUA	FUNA	IMSU	ESUST	ASU	MUO	Total
Academic	1,450	1,400	1205	1,200	1,100	1.050	650	400	300	8,755
Registry	60	55	52	45	42	40	35	30	30	389
Total	1,510	1,455	1,257	1,245	1,142	1,090	685	430	330	9,144

Table 5: Population (N) Definition Based on Selected Nigerian Universities
(Field work)

STAFF	UNN	NAU	FUTO	MOUA	FUNA	IMSU	ESUST	ASU	MUO	Total
Academic	252	248	235	234	225	221	176	127	108	1,826
Registry	28	26	25	20	20	19	17	14	14	183
Total	280	274	260	254	245	240	193	141	122	2,009

Table 3.5: Sample Population (n) Definition Based on Selected Nigerian Universities
(Field work)

As established by Gay and Diehl (1992), (as cited in Hashim, 2010) and supported by Hill (1998), 30 respondents per group is the minimum needed for correlational and experimental research. This left us with the minimum population sample of 270 respondents. Also, as suggested by Baltar & Brunet (2012), the researcher used the stratified random sampling technique to select class advisers and result processors, result vetters, and transcript processors, as the best-fit respondents in matters of students' results production.

3.3.2. Primary Data

Our interest in this study is to find whether there is a significant difference between the Existing System and the proposed HRPMS system. A study of this nature requires primary data obtained from a survey since such data was not recorded anywhere. Hence, the data used was collected and compiled by the researcher himself. Stratified random sampling was used to draw a sample of 360 respondents from the population, and questionnaires were distributed to them. Out of the 360 questionnaires distributed by two-round Delphi technique, 290 were duly returned and used. This represents 81 % returns.

3.3.3. Questionnaire Design and Validation

Following the standard for questionnaire design (Ritchie et al., 2013; Barua, 2013; Murray, 2013), a Likert summated five-point scale questionnaire was designed for data collection with the performance variables. To validate the questionnaire design as opined by Csikszentmihalyi and Larson (2014), it was subjected to vetting by IT professionals and statisticians; to ensure that its structure, framing, content and language reflect the study objectives. The questionnaire reliability was done using Cronbach Alpha Technique for establishing its internal consistency, having rated each response accordingly. The researcher administered the instrument to 30 respondents from Federal Polytechnic Nekede Owerri, outside the research area of this study, and the reliability coefficient (α **)** of the questionnaire instrument was calculated using Equation 1:

$$\alpha = \frac{N\bar{c}}{\bar{v} + (N-1)\bar{c}} \quad \text{...} (1)$$

Where: N = number of respondents; $\bar{c}$ **=** average inter-item covariance among respondents; and $\bar{V}$ = average variance among respondents. For the calculation, N= 30; $\bar{c}$ **=** 0.124; $\bar{V}$ = 0.8027; hence, $\alpha \cong$ 0.85. Since, instruments with reliability index of 0.60 and above is said to be reliable (Nkwocha, 2010); the questionnaire can be said to be reliable. This validates the questionnaire both in framing and content.

3.4. Hrpms Statistical Validation Method: Multivariate Linear System Analysis

In the fourth Agile test quadrant, the statistical analysis of performance was done using the multivariate linear system model analysis to test the hypotheses and predict the performance difference verifying the HRPMS performance improvement. Based on the same variables (X_1, X_2, X_3, X_4, X_5 and X_6), two data samples were collected from the selected Universities by two-round Delphi technique. The first sample of the variates need to be analyzed at the same time, and their effects compared simultaneously with the second sample that has the same level. Therefore, linear multivariate system analysis provides the best model for this problem (Chukwu, 2007); which involves Hotelling's T^2 analysis, Mahalanobi's D^2 analysis and determination of the F-Distribution. Hence, these analyses were carried out using MATLAB and Excel packages as computation tools; to test the null hypothesis that there is no significant difference in performance between the Existing System and the proposed HRPMS against the alternative hypothesis that HRPMS performs better. The test statistics are stated in Equation 2 as follows:

$$T^2 = \frac{n_1 n_2}{n_1 + n_2} D^2 \quad \text{...........} (2)$$

Where: T^2 is the Hotelling's T^2 distribution; n_1 is the number of observations in the sample data for the Existing system; n_2 is the number of observations in the sample data for the HRPMS, and D^2 is the Mahalanobi's D^2 statistic, given by Equation 3, as follows:

$$D^2 = \left(\overline{X}_1 - \overline{X}_2\right)^T S^{-1}\left(\overline{X}_1 - \overline{X}_2\right) \quad \text{...........} (3)$$

Where: $\overline{X}_1$ denote the mean of the existing system (column vector); $\overline{X}_2$ denote the mean of the HRPMS (column vector); $\left(\overline{X}_1 - \overline{X}_2\right)$ is the difference in mean of the Existing system and that of the HRPMS (a column vector); $\left(\overline{X}_1 - \overline{X}_2\right)^T$ is the transpose of the difference in mean of both systems (row vector); S stands for the variance-covariance matrix or dispersion matrix or information matrix of the combined sample from the two (2) separate data samples of the existing system and the HRPMS; and S^{-1} denote the inverse of information matrix.

The mean vector $\overline{X}_i$ of the individual samples i **;** for the six (6) variables under consideration is given by Equation 4, as follows:

$$\overline{X}_i = \begin{bmatrix} {}_i\overline{X}_1 \\ {}_i\overline{X}_2 \\ {}_i\overline{X}_3 \\ {}_i\overline{X}_4 \\ {}_i\overline{X}_5 \\ {}_i\overline{X}_6 \end{bmatrix} \quad \text{...........} (4)$$

Where i = 1 and 2 (for existing system and HRPMS data samples respectively); and ${}_i\overline{X}_1 = (\sum_1^{290} {}_iX_1)/290$; ${}_i\overline{X}_2 = (\sum_1^{290} {}_iX_2)/290$; ${}_i\overline{X}_3 = (\sum_1^{290} {}_iX_3)/290$; ${}_i\overline{X}_4 = (\sum_1^{290} {}_iX_4)/290$; ${}_i\overline{X}_5 = (\sum_1^{290} {}_iX_5)/290$; and ${}_i\overline{X}_6 = (\sum_1^{290} {}_iX_6)/290$, representing each of the six (6) variables under consideration. Where: X_1 = Usefulness, X_2 = Ease of Access and Use, X_3 = Security and Confidentiality, X_4 = Processing Speed, X_5 = Computation Accuracy, and X_6 = System Performance.

Laying out the variance-covariance (information) matrix for the separate and combined data samples containing the six (6) variables under consideration, we have Equation 5 as follows:

$$(n_i - 1)S_i^2 = \begin{bmatrix} X_{11} & X_{12} & X_{13} & X_{14} & X_{15} & X_{16} \\ X_{21} & X_{22} & X_{23} & X_{24} & X_{25} & X_{26} \\ X_{31} & X_{32} & X_{33} & X_{34} & X_{35} & X_{36} \\ X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} \\ X_{51} & X_{52} & X_{53} & X_{54} & X_{55} & X_{56} \\ X_{61} & X_{62} & X_{63} & X_{64} & X_{65} & X_{66} \end{bmatrix} \quad \text{...........} (5)$$

Where n_i and S_i are as earlier defined; and the variances are:

$X_{11} = \sum_1^{290} X_1^2 - n_i\overline{X}_1^2$; $X_{22} = \sum_1^{290} X_2^2 - n_i\overline{X}_2^2$; $X_{33} = \sum_1^{290} X_3^2 - n_i\overline{X}_3^2$; $X_{44} = \sum_1^{290} X_4^2 - n_i\overline{X}_4^2$; $X_{55} = \sum_1^{290} X_5^2 - n_i\overline{X}_5^2$; $X_{66} = \sum_1^{290} X_6^2 - n_i\overline{X}_6^2$;

and the covariances are: $X_{12} = \sum_1^{290} X_1X_2 - n_i\overline{X}_1\overline{X}_2$; $X_{13} = \sum_1^{290} X_1X_3 - n_i\overline{X}_1\overline{X}_3$; $X_{14} = \sum_1^{290} X_1X_4 - n_i\overline{X}_1\overline{X}_4$;

$X_{15} = \sum_1^{290} X_1X_5 - n_i\overline{X}_1\overline{X}_5$; $X_{23} = \sum_1^{290} X_2X_3 - n_i\overline{X}_2\overline{X}_3$; $X_{24} = \sum_1^{290} X_2X_4 - n_i\overline{X}_2\overline{X}_4$;

$X_{25} = \sum_1^{290} X_2X_5 - n_i\overline{X}_2\overline{X}_5$; $X_{26} = \sum_1^{290} X_2X_6 - n_i\overline{X}_2\overline{X}_6$; $X_{34} = \sum_1^{290} X_3X_4 - n_i\overline{X}_3\overline{X}_4$;

$X_{35} = \sum_{1}^{290} X_3 X_5 - n_i \overline{X}_3 \overline{X}_5$; $X_{36} = \sum_{1}^{290} X_3 X_6 - n_i \overline{X}_3 \overline{X}_6$; $X_{45} = \sum_{1}^{290} X_4 X_5 - n_i \overline{X}_4 \overline{X}_5$;

$X_{46} = \sum_{1}^{290} X_4 X_6 - n_i \overline{X}_4 \overline{X}_6$; $X_{56} = \sum_{1}^{290} X_5 X_6 - n_i \overline{X}_5 \overline{X}_6$;

and the symmetric covariances are: $X_{21}=X_{12}$; $X_{31}=X_{13}$; $X_{32}=X_{23}$; $X_{41}=X_{14}$; $X_{42}=X_{24}$; $X_{43}=X_{34}$; $X_{51}=X_{15}$; $X_{52}=X_{25}$; $X_{53}=X_{35}$; $X_{54}=X_{45}$; $X_{61}=X_{16}$; $X_{62}=X_{26}$; $X_{63}=X_{36}$; $X_{64}=X_{46}$; $X_{65}=X_{56}$.

However, pooling the two layout matrices from the separate data samples will give a combined layout matrix given by equation 6, thus:

$$(n_1 + n_2 - 2)S = \begin{bmatrix} \sum X_{11} & \sum X_{12} & \sum X_{13} & \sum X_{14} & \sum X_{15} & \sum X_{16} \\ \sum X_{21} & \sum X_{22} & \sum X_{23} & \sum X_{24} & \sum X_{25} & \sum X_{26} \\ \sum X_{31} & \sum X_{32} & \sum X_{33} & \sum X_{34} & \sum X_{35} & \sum X_{36} \\ \sum X_{41} & \sum X_{42} & \sum X_{43} & \sum X_{44} & \sum X_{45} & \sum X_{46} \\ \sum X_{51} & \sum X_{52} & \sum X_{53} & \sum X_{54} & \sum X_{55} & \sum X_{56} \\ \sum X_{61} & \sum X_{62} & \sum X_{63} & \sum X_{64} & \sum X_{65} & \sum X_{66} \end{bmatrix} \qquad (6)$$

Where the elements $X_{qr}; q = 1,...,6; r = 1,...,6$ in the same position in the layout matrices $(n_1 - 1)S_1^2$ and $(n_2 - 1)S_2^2$ of samples 1 and 2 are added to get the elements X_{qr} of the pooled $(n_1 + n_2 - 2)S$ layout matrix; while all other parameters retain their definitions.

From the combined layout matrix $(n_1 + n_2 - 2)S$, the variance-covariance (information) matrix S is determined as given in Equation 7, thus:

$$S = \begin{bmatrix} \sum X_{11} & \sum X_{12} & \sum X_{13} & \sum X_{14} & \sum X_{15} & \sum X_{16} \\ \sum X_{21} & \sum X_{22} & \sum X_{23} & \sum X_{24} & \sum X_{25} & \sum X_{26} \\ \sum X_{31} & \sum X_{32} & \sum X_{33} & \sum X_{34} & \sum X_{35} & \sum X_{36} \\ \sum X_{41} & \sum X_{42} & \sum X_{43} & \sum X_{44} & \sum X_{45} & \sum X_{46} \\ \sum X_{51} & \sum X_{52} & \sum X_{53} & \sum X_{54} & \sum X_{55} & \sum X_{56} \\ \sum X_{61} & \sum X_{62} & \sum X_{63} & \sum X_{64} & \sum X_{65} & \sum X_{66} \end{bmatrix} / (n_1 + n_2 - 2) \qquad (7)$$

To test hypothesis one, the probability distribution function of the pooled sample was fitted, and inference was drawn from it. By fitting the generalized multivariate Probability Distribution Function (PDF), we obtained Equation 8 as the system performance density estimation model, as follows:

$$F(x_q, \mu, v) = (2\pi)^{-p/2} \, |S|^{-1/2} \exp[-1/2(\mu)^T S^{-1}(\mu)] \qquad (8)$$

Where: $(x_q)_{q=1,...,p}$ are the variables under consideration; p denotes the number of variates; $\mu = (\overline{X}_1 - \overline{X}_2)$ is the difference in mean of the two data samples matrices $[S_1]$ and $[S_2]$ of the existing system and HRPMS; $v = S$ stands for the information matrix from the pooled or combined sample of $[S_1]$ and $[S_2]$ data; $|S|$ is the determinant of the information matrix; and the other notations retain their usual meanings as earlier defined.

This PDF statistical model predicts the density or effect of the non-functional user requirement variables on average performance of the information systems (existing and proposed), given that their data samples $[S_1]$ and $[S_2]$ are based on the same performance variables ($x_q; q=1,...,6;\ p = 6$). Hence, hypothesis one is given in Equation 9 as follows:

H_{O1}: $F(x_q, \mu, v) = 0$ VS H_{A1}: $F(x_q, \mu, v) \neq 0$ .... (9)

Where: H_{O1}: stands for the null hypothesis one, which says that the non-functional variables have no statistical and significant effect on system performance; H_{A1}: stands for the alternate hypothesis one, which says that the non-functional variables have statistical and significant effect on system performance; "VS" denotes contrast; other notations retain their definitions. Therefore, the decision rule is such that null hypothesis one (H_{01}) is accepted if: $F(x_q, \mu, v) = 0$, otherwise, reject H_{01} and accept the alternate hypothesis one (H_{A1}); and we conclude that the non-functional variables have statistical and significant effect on system performance – positive or negative. Positive effect means direct effect and negative effect means indirect effect. When Equation 9 tends to one (i.e. $F(x_q, \mu, v) => \pm 1$), there exists a strong effect, but when it tends to zero (i.e. $F(x_q, \mu, v) => \pm 0$), there exists a weak effect.

To test hypothesis two, the F-distribution for this work was determined. The F-calculated (**F_{cal}**) and the F-tabulated (**F_{tab}**) of the pooled data sample, are given in Equations 10 and 11 as follows:

$$F_{cal} = \frac{(n_1 + n_2 - p - 1)}{p(n_1 + n_2 - 2)}.T^2 \quad \text{......} (10)$$

and $$F_{tab} = F_{p,(n_1+n_2-p-1)}(\alpha) \quad \text{......} (11)$$

Where: **F_{tab}** is F value in standard F-distribution table; (α) denote the level of significance; $p \,\&\, (n_1 + n_2 - p - 1)$ denote the degrees of freedom; other notations are as defined earlier. In this study, hypothesis two was tested at 5% level of significance, the correlation on the performance of the two systems (existing and proposed) was determined, and inference was drawn. Hypothesis two is given in Equation 12 as follows:

H_{O2}: $\overline{X}_1 = \overline{X}_2$ VS H_{A2}: $\overline{X}_1 \neq \overline{X}_2$ ...... (12)

Where: H_{O2}: stands for the null hypothesis two, which says that there is no significant difference in the mean performance of the Existing System and HRPMS; H_{A2}: stands for the alternate hypothesis two, which says that there is significant difference in the mean performance of the Existing System and HRPMS; "VS" denotes contrast; other notations retain their earlier definitions.

To make inference and conclusion on hypothesis two, the decision rule in equation 13 says: Accept Null Hypothesis one (H_{O1}) if:

$$F_{cal} < F_{tab} \text{ -......} (13)$$

Otherwise, reject (H_{01}) and accept the alternative hypothesis one (H_{A1}); and conclude that there is significant difference between the mean performances of both systems. Therefore, we state that the HRPMS performs better.

4. Results and Discussion

This section presents the HRPMS performance validation and hypothesis testing results following multivariate analysis of data collected from tester respondents. It also discusses the findings made in this work in terms of system and user non-functional requirements analysis of HRPMS performance.

4.1. HRPMS Performance Validation and Hypothesis Testing

This analysis was done in the fourth or technology-facing product critique Agile test quadrant through multivariate linear system statistical analysis and test of hypotheses. From the beta test stage, the multivariate linear system analysis of the HRPMS performance variables was done using: X_1 = Usefulness, X_2 = Ease of Access and Use, X_3 = Security and Confidentiality, X_4 = Processing Speed, X_5 = Computation Accuracy, X_6 = System Performance, and n_1 = 290 observations. The primary data sample obtained from the first round of the Delphi technique for the existing system (**see** VI), represents 81% questionnaire returns. Thus, for i = 1, the mean vector ($\overline{X}_1$) of the existing system becomes:

$$\overline{X}_1 = \begin{bmatrix} {}_1\overline{X}_1 \\ {}_1\overline{X}_2 \\ {}_1\overline{X}_3 \\ {}_1\overline{X}_4 \\ {}_1\overline{X}_5 \\ {}_1\overline{X}_6 \end{bmatrix} = \begin{bmatrix} 14.77509 \\ 14.77855 \\ 14.77163 \\ 14.37370 \\ 14.95843 \\ 14.03460 \end{bmatrix}; \text{And}$$

The layout matrix $(n_i - 1)S_i^2$ of the Existing system becomes:

$$(n_1 - 1)S_1^2 = \begin{bmatrix} 1788 & 350.2222 & 353.8728 & 272.9138 & -572.5234 & -508.1187 \\ 350.2222 & 1745.4 & 1310.1 & 1058.5 & -477.5326 & -266.201 \\ 353.8728 & 1310.1 & 1784.7 & 1079.3 & -475.5141 & -268.0363 \\ 272.9138 & 1058.5 & 1079.3 & 1317.1 & -469.3157 & -314.4477 \\ -572.5234 & -477.5326 & -475.5141 & -469.3157 & 2140.2 & -849.3187 \\ -508.1187 & -266.201 & -268.0363 & -314.4477 & -849.3187 & 974.7008 \end{bmatrix}$$

Again, with the same performance variables (X_1, X_2, X_3, X_4, X_5, X_6), and n_2 = 290 observations, another primary data sample was obtained from the second round of the Delphi technique for the HRPMS, and it also represents 81% questionnaire returns. Thus, for i = 2, the mean vector ($\overline{X}_1$) of the HRPMS becomes:

$$\overline{X}_2 = \begin{bmatrix} {}_2\overline{X}_1 \\ {}_2\overline{X}_2 \\ {}_2\overline{X}_3 \\ {}_2\overline{X}_4 \\ {}_2\overline{X}_5 \\ {}_2\overline{X}_6 \end{bmatrix} = \begin{bmatrix} 15.40138 \\ 17.43945 \\ 17.51557 \\ 16.86851 \\ 17.26644 \\ 17.22145 \end{bmatrix}$$; And

The layout matrix $(n_i - 1)S_i^{\,2}$ of the HRPMS becomes:

$$(n_2 - 1)S_2^{\,2} = \begin{bmatrix} 2798.3 & -63.5630 & -61.5454 & -17.5164 & 1130.2 & -361.8877 \\ -63.5630 & 309.0193 & 254.0469 & 145.5144 & -51.9929 & -114.4587 \\ -61.5454 & 254.0469 & 245.3942 & 105.1454 & -55.1462 & -126.6188 \\ -17.5164 & 145.5144 & 105.1454 & 52.4774 & -56.1436 & -173.0584 \\ 1130.2 & -51.9929 & -55.1462 & -56.1436 & 2422.3 & -73.4086 \\ -361.8877 & -114.4587 & -126.6188 & -173.0584 & -73.4086 & 455.2814 \end{bmatrix}$$

The pooled (combined) data sample layout matrix $(n_1 + n_2 - 2)S$ for both the Existing system and the HRPMS with n_1 = n_2 = 290 observations becomes:

$$(n_1 + n_2 - 2)S = \begin{bmatrix} 4586.3 & 286.659 & 292.327 & 255.397 & 557.677 & -870.006 \\ 286.659 & 2054.419 & 1564.147 & 1204.014 & -529.526 & -380.66 \\ 292.327 & 1564.147 & 2030.094 & 1184.445 & -530.66 & -394.655 \\ 255.397 & 1204.014 & 1184.445 & 1369.577 & -525.459 & -487.506 \\ 557.677 & -529.526 & -530.66 & -525.459 & 4562.5 & -922.727 \\ -870.006 & -380.66 & -394.655 & -487.506 & -922.727 & 1429.982 \end{bmatrix}$$

The variance-covariance (information) matrix S for the pooled sample is thus:

$$S = \begin{bmatrix} 7.9348 & 0.4960 & 0.5058 & 0.4419 & 0.9648 & -1.5052 \\ 0.4960 & 3.5544 & 2.7061 & 2.0831 & -0.9161 & -0.6586 \\ 0.5058 & 2.7061 & 3.5123 & 2.0492 & -0.9181 & -0.6828 \\ 0.4419 & 2.0831 & 2.0492 & 2.3695 & -0.9091 & -0.8434 \\ 0.9648 & -0.9161 & -0.9181 & -0.9091 & 7.8936 & -1.5964 \\ -1.5052 & -0.6586 & -0.6828 & -0.8434 & -1.5964 & 2.4740 \end{bmatrix}$$

The inverse S^{-1} of the information matrix S becomes:

$$S^{-1} = \begin{bmatrix} 0.1429 & -0.00911 & -0.00774 & 0.02061 & 0.001185 & 0.09017 \\ -0.00911 & 0.7998 & -0.41447 & -0.35306 & -0.00054 & -0.02773 \\ -0.00774 & -0.41447 & 0.79136 & -0.31435 & 0.00910 & 0.00206 \\ 0.02061 & -0.35306 & -0.31435 & 1.14866 & 0.11203 & 0.29569 \\ 0.001185 & -0.00054 & 0.00910 & 0.11203 & 0.17112 & 0.15170 \\ 0.09017 & -0.02773 & 0.00206 & 0.29569 & 0.15170 & 0.65094 \end{bmatrix}$$

And, the determinant $|S|$ of the information matrix S gives: $|S| = 458.3894$.

Also, the difference in mean vector $(\overline{X}_1 - \overline{X}_2)$ of the two data samples become:

$$(\overline{X}_1 - \overline{X}_2) = \begin{bmatrix} -0.62629 \\ -2.66090 \\ -2.74394 \\ -2.49481 \\ -2.30801 \\ -3.18685 \end{bmatrix}$$; and

$$(\overline{X}_1 - \overline{X}_2)^T = [-0.62629 \;\; -2.66090 \;\; -2.74394 \;\; -2.49481 \;\; -2.30801 \;\; -3.18685]$$

4.1.1. Testing Hypothesis One

Fitting the Probability Distribution Function (PDF) density estimation model for this study gives:

$$F(x_1,..., x_6, \mu, v) = (2\pi)^{-3} |S|^{-1/2} \exp[-1/2(\mu)^T S^{-1}(\mu)] = 1.043458X10^{-8}$$

Where: $-\infty < x < \infty$; $x_q; q = 1,...,6$; $p = 6$; $\mu = (\overline{X}_1 - \overline{X}_2) > 0$; $v = S$; $|S| = 458.3894$;

$$(\overline{X}_1 - \overline{X}_2)^T = [-0.62629 \; -2.66090 \; -2.74394 \; -2.49481 \; -2.30801 \; -3.18685];$$

$$S^{-1} = \begin{bmatrix} 0.1429 & -0.00911 & -0.00774 & 0.02061 & 0.001185 & 0.09017 \\ -0.00911 & 0.7998 & -0.41447 & -0.35306 & -0.00054 & -0.02773 \\ -0.00774 & -0.41447 & 0.79136 & -0.31435 & 0.00910 & 0.00206 \\ 0.02061 & -0.35306 & -0.31435 & 1.14866 & 0.11203 & 0.29569 \\ 0.001185 & -0.00054 & 0.00910 & 0.11203 & 0.17112 & 0.15170 \\ 0.09017 & -0.02773 & 0.00206 & 0.29569 & 0.15170 & 0.65094 \end{bmatrix}; \text{and}$$

$$(\overline{X}_1 - \overline{X}_2) = \begin{bmatrix} -0.62629 \\ -2.66090 \\ -2.74394 \\ -2.49481 \\ -2.30801 \\ -3.18685 \end{bmatrix}$$

Since: $F(x_q, \mu, v) = 1.043458\ X10^{-8} \neq 0$, we reject $H_{O1} : F(x_q, \mu, v) = 0$ and accept $H_{A1} : F(x_q, \mu, v) \neq 0$; and conclude that the non-functional variable have statistical and significant effect on system performance - though positive (direct) but slightly weak. Consequently, this confirms that the non-functional requirement variables actually drive system performance.

Again, $PDF = F(x_q, \mu, v) = (2\pi)^{-p/2} |S|^{-1/2} \exp[-1/2(\mu)^T S^{-1}(\mu)] = 1.043458X10^{-8}$ is a prediction of the performance difference (P_D) of the HRPMS (from the existing system) based on a set of performance variables ($x_q; q = 1,..., p; p > 1$) with dual-sample data (S_1 and S_2). The predicted P_D value (1.043458X10^{-8}); shows a positively performing (or improved) HRPMS. This verifies the HRPMS non-functional requirements and system performance improvement statistically.

4.1.2.Testing Hypothesis Two

Following from the already found and defined statistics and values, which includes: difference in mean $(\overline{X}_1 - \overline{X}_2)$, inverse of the information matrix (S^{-1}), the difference in mean transpose $(\overline{X}_1 - \overline{X}_2)^T$, n_1, n_2, p and α ; Mahalanobi's D^2 statistic gives: $D^2 = (\overline{X}_1 - \overline{X}_2)^T S^{-1} (\overline{X}_1 - \overline{X}_2) = 19.6013$; and Hotelling's T^2 statistic gives:

$$T^2 = \frac{n_1 n_2}{n_1 + n_2} D^2 = \frac{290 \times 290}{580}(19.6013) = 2842.1885$$

Thus, the calculated F statistic (F_{cal}) gives:

$$F_{cal} = \frac{n_1 + n_2 - p - 1}{p(n_1 + n_2 - 2)} . T^2 = \frac{290 + 290 - 6 - 1}{6(290 + 290 - 2)} \times 2842.1885 = 469.6004$$

Also, the Tabulated F statistic (F_{tab}) gives:

$$F_{tab} = F_{p,(n_1+n_2-p-1)}(\alpha) = F_{6,573}(0.05) = F_{6,600}(0.05) = 2.50$$

Comparing $F_{cal} = 469.6004$ and $F_{tab} = 2.50$, we see that $F_{cal} > F_{tab}$; hence we reject $H_{O2} : \overline{X}_1 = \overline{X}_2$; and accept $H_{A2} : \overline{X}_1 \neq \overline{X}_2$; and conclude that there is significant difference between the mean performances of the single-platform system and the HRPMS. Hence, the HRPMS performs better than the existing single-platform result system.

5. Discussion of Findings

The findings made in the course of this work as discussed, are as follows:

- The existing single-platform result systems were analyzed using the qualitative survey on the performance variables. The result showed that the standalone system is strong with usefulness, security and confidentiality, and computation accuracy; and weak with ease of access and use online, and processing speed. These represents strength-to-weakness ratio of 10:8, representing sixty percent (60%) strength and forty percent (40%) deficiency. Also, the client-server (network) system is strong with usefulness and processing speed; and weak with ease of use, security and confidentiality, and computation accuracy. These represents strength-to-weakness ratio of 9:9, representing fifty percent (50%) strength and fifty percent (50%) deficiency. Hence, none of the two can singly carry out result processing and management adequately, but can as a hybrid.
- The HRPMS framework was conceptualized, and the outcome was a hybrid (cross-platform) model to optimize and improve the deficiencies of the single platform model.
- The HRPMS logical structure was designed using the UML diagrams, producing high level models, flowcharts, use case, class, and entity-relationship models of the system.
- The HRPMS module designs were coded and configured using Excel VBA 2013, Java 8.0, PHPDesigner 8.0 and MySQL DBMS 5.4 program application platforms, which produced usable standalone and client-server modules input-output forms.
- Fitting system PDF gave a predicted value of $1.043458X10^{-8}$ which shows a positively performing (or improved) HRPMS.
- Testing hypothesis one (1) shows that: $F(x_q, \mu, v) = 1.043458\ X10^{-8} \neq 0$, so we rejected $H_{O1} : F(x_q, \mu, v) = 0$ and accepted $H_{A1} : F(x_q, \mu, v) \neq 0$; and concluded that the non-functional variables have significant positive (direct) but slightly weak effect on system performance. Thus, confirming that the non-functional requirement variables actually drives system performance.
- Also, testing hypothesis two (2) compares $F_{cal} = 469.6004$ and $F_{tab} = 2.50$, and we see that $F_{cal} > F_{tab}$; hence $H_{O2} : \overline{X}_1 = \overline{X}_2$ was rejected and $H_{A2} : \overline{X}_1 \neq \overline{X}_2$ was accepted; and we concluded that there is significant difference between the mean performances of the existing single-platform system and the HRPMS. Thus, the HRPMS performs better.

6. Conclusions

Motivated by the problem statement that the available single-platform system models are seriously deficient and inadequate to drive the NUC's objective of attaining higher global ranking and competitiveness, this work set out to develop a hybrid result processing and management system for Nigerian Universities. The study established by survey the performance criteria for analyzing the performances of the existing and proposed systems, modelled the HRPMS framework that optimizes the standalone-network hybrid functionalities, designed the logical and physical structures of the HRPMS input-output modules using UML diagrams, coded and configured the HRPMS applications using Excel Binary VBA Programming, Java 8.0 Programming, and MySQL5.4 DBMS, and validated the HRPMS operation and performance statistically. Conclusively, the hybrid system provides an efficient and more secured model for students' result processing and management in Nigerian Universities. This model can stimulate the higher ranking of the Nigerian Universities for global competitiveness. The HRPMS processing and management is inherently distributed and produces redundant (hard-to-crash) outputs that can always be compared where necessary. This reduces to a minimum the risk of total system failure and total data lose when systems fail. It can also aid data recovery in case of system or database compromise or data corruption. The benefit of the HRPMS model is that it provides for a more secured and much improved process automation of result processing and management.

7. References

i. Abran, A., Moore, J. W., Bourque, P., Dupuis, R., & Tripp, L. L. (2004). Software Engineering Body of Knowledge. IEEE Computer Society, Angela Burgess, 15-50.
ii. Agard, B. & Kusiak A. (2004, August). Standardization of Components, Products and Processes with Data Mining. International Conference on Production Research Americas 2004, Santiago, Chile, 1- 4.
iii. Agile Alliance (2001, February). Manifesto for Agile Software Development. 1-70, https://www.agilealliance.org/agilemanifesto
iv. Ahmed M. Z. (2010). Basic Concepts of UML. Lecture notes on Unified Modeling Language, MCA III, 1-20.
v. Akinmosin, J. (2014). Automated Students Result Management System Using Oracle's Database, Forms and Reports. Journal of Information Engineering and Applications, 4 (11), 1-11, ISSN 2224-5782 (print), ISSN 2225-0506 (online), www.iiste.org.
vi. Anyiam, F. N. (2017). Enhanced Result Processing and Checking System for Universities in Imo state. (Unpublished Work).
vii. Asproth, V. (2012). Information technology challenges for long-term preservation of electronic information, International Journal of Public Information Systems, 1 (1), 20.
viii. Aveling, B. (2004). XP Lite Considered Harmful? Proceedings of the 5th International Conference of Extreme Programming and Agile Processes in Software Engineering, Springer, LNCS 3092, 94-103.

ix. Avison, D., Lau, F., Myers, M., & Nielsen, P. (1999, January). Action Research. Communications of the ACM, 42(1), 94-97.
x. Bahill, A. T., & Madni, A. M. (2017). Discovering system requirements in Tradeoff Decisions in System Design. Springer International, 373-457.
xi. Baltar, F., & Brunet, I. (2012). Social research 2.0: virtual snowball sampling method using Facebook. Internet Research, 22(1), 57-74.
xii. Barua, A. (2013). Methods for decision-making in survey questionnaires based on Likert scale. Journal of Asian Scientific Research, 3 (1), 35.
xiii. Battiston, S., Farmer, J. D., Flache, A., Garlaschelli, D., Haldane, A. G., Heesterbeek, H., & Scheffer, M. (2016). Complexity theory and financial regulation. Science, 351(6275), 818-819.
xiv. Beka, A. P. & Beka, F. T. (2015, September). Automated Result Processing System: A Case Study of Nigerian University. International Journal for Research in Emerging Science and Technology (IJREST), 2 (9), 51-68, E-ISSN: 2349-7610.
xv. Ben-Arieh, P., & Qian, L. (2003). Activity-based Cost Management for Design and Development Stage. Int. J. Prod. Econ., 83 (2), 169-183.
xvi. Bhat, J. M., Gupta, M., & Murthy, S. N. (2006). Overcoming requirements engineering challenges: Lessons from offshore outsourcing. IEEE software, 23 (5), 38-44.
xvii. Boegh, J. (2008). A new standard for quality requirements. IEEE Software, 25 (2), 57.
xviii. Boehm, B., & Turner, R. (2005). Management challenges to implement agile processes in traditional development organizations. IEEE Software, 22 (5), 30-40.
xix. Bolu, C., Adewumi, A. O., & Egbo, K. (2014, February). Towards Making World-Class Universities: Case Study of the role of Information and Communication Technology. International Journal of Advanced Research in IT and Engineering, 3(2), 9-25. ISSN: 2278-6244, www.garph.co.uk.
xx. Boons, F., & Lüdeke-Freund, F. (2013). Business models for sustainable innovation: state-of-the-art and steps towards a research agenda. Journal of Cleaner Production, 45, 9-19.
xxi. Chettri, B., Prasad, S. K., Sharma, B., & Manger, N. (2016, April). Accessing a Portion of MIS: Result Management, International Journal of Engineering Trends and Technology (IJETT), 34 (5), 205-208, ISSN: 2231-5381, http://www.ijettjournal.org.
xxii. Chukwu, B. A. (2007). Panel data analysis of determinants of trade in Africa and Asia. Nigerian journal of economic and social studies, 49 (3), 1-15.
xxiii. Cover, T. M., & Thomas, J. A. (2012). Elements of information theory. John Wiley & Sons, 15-20.
xxiv. Creswell, J. W. (2013). Research design: Qualitative, quantitative, and mixed methods approaches. Sage publications, 22-34.
xxv. Csikszentmihalyi, M., & Larson, R. (2014). Validity and reliability of the experience-sampling method, in flow and the foundations of positive psychology. Springer Netherlands, 35-54.
xxvi. Dabbagh, M., & Lee, S. P. (2014). An approach for integrating the prioritization of functional and nonfunctional requirements. The Scientific World Journal, 14.
xxvii. DaSilva, I. F., Neto, P. A. D. M. S., O'Leary, P., de Almeida, E. S., & de LemosMeira, S. R. (2015). Using a multi-method approach to understand Agile software product lines. Information and Software Technology, 57, 527-542.
xxviii. Deemer, P., Benefield, G., Larman, C., & Vodde, B. (2012). The Scrum Primer: A Lightweight Guide to the Theory and Practice of Scrum, Version 2.0. 3-16. http://www.infoq.com/minibooks/Scrum_Primer.
xxix. Demirkan, H., & Delen, D. (2013). Leveraging the capabilities of service-oriented decision support systems: Putting analytics and big data in cloud. Decision Support Systems, 55 (1), 412-421.
xxx. Desouza, K. (2003, June). Facilitating Tacit Knowledge Exchange, Communications of the ACM, 46 (6), 85-88.
xxxi. Douglas, D. & Glen, A. (2000). Planning Extreme Programming. Reading, MA: Addison-Wosley, 20-45.
xxxii. Eckhardt, J., Vogelsang, A., & Fernández, D. M. (2016, May). Are non-functional requirements really non-functional?: an investigation of non-functional requirements in practice. In Proceedings of the 38th International Conference on Software Engineering, ACM, 832-842.
xxxiii. Eke, B. O., & Nweke, O. E. (2016). An Efficient Cloud-Based Result Security System Using Digital Certificate. International Journal, 6 (3), 1-10.
xxxiv. Eklund, U., & Bosch, J. (2013, September). Archetypical approaches of fast software development and slow embedded projects. In 2013 39th Euromicro Conference on Software Engineering and Advanced Applications, IEEE, 276-283.
xxxv. Eludire, A. A. (2011, August). The Design and Implementation of Student Academic Record Management System. Research Journal of Applied Sciences, Engineering and Technology, Maxwell Scientific Organization, 3 (8), 707-712, ISSN: 2040-7467.
xxxvi. Esawi, A., & Ashby, M. F. (2003). Cost Estimates to guide Pre-selection of Processes. Mater. Des. 24, (8), 605-616, doi: 10.1016/50261-3069(03)00136-6.
xxxvii. Esiefarienrhe B. M., Adeiza, M. R., & Ejura, A. G. (2015). Development of Commuter Booking Software for Road Transport Company. Development, 120 (19), 10-17.
xxxviii. Ferro, E., Loukis, E. N., Charalabidis, Y., & Osella, M. (2013). Policy making 2.0: From theory to practice. Government Information Quarterly, 30 (4), 359-368.

xxxix. Fowler, M., & Highsmith, J. (2014). The agile manifesto (2001), 1-25, www. pmp-projects. org/Agile-Manifesto. pdf.

xl. Graham, C. R., Woodfield, W., & Harrison, J. B. (2013). A framework for institutional adoption and implementation of blended learning in higher education. The internet and higher education, 18, 4-14.

xli. Griffiths, P. E., & Tabery, J. (2013). Developmental systems theory: What does it explain, and how does it explain it. Adv Child Dev Behav, 44, 65-94.

xlii. Grossman, T. A. (2006, September). Integrating Spreadsheet Engineering in a Management Science Course - A Hierarchical Approach". INFORMS Transactions on Education 7(1), 18-36, http://www.informs.org/Journal/ITE/ Archive/Volume-7/Integrating Spreadsheet-Engineering-in-a-Management-Science-Course-A-Hierarchical-Aproach

xliii. Grossman, T. A., & Özlük, Ö. (2011). Spreadsheets Grow Up: Three Spreadsheet Engineering Methodologies for Large Financial Planning Models. San Francisco CA, 117-145.

xliv. Grossman, T. A., Mehrotra, V. & Özlük Ö. (2007, December). Lessons from Mission-Critical Spreadsheets, Communications of the Association for Information Systems 20 (60), 1009-1042.

xlv. Hashim, Y., A. (2010). Determining Sufficiency of Sample Size in Management Survey Research Activities. International Journal of Organizational Management & Entrepreneurship Development, 6 (1), 119-130.

xlvi. Hastie, S., & Wojewoda, S. (2015). Standish Group 2015 Chaos Report-Q & A with Jennifer Lynch, 1-25.

xlvii. Hemmatian, H., Fereidoon, A., Sadollah, A., & Bahreininejad, A. (2013). Optimization of laminate stacking sequence for minimizing weight & cost using elitist & system optimization. Advances in engineering Software, 57, 8-18.

xlviii. Herzberg, B., & Sisombat, L. (2013). An Attempt to Develop a Global Indicator on Private-Sector Involvement in Public Policies and Strategies, State-Business Relations and Industrial Policy: Current Policy and Research Debates. London: Growth Research Prog., 1-25, https://static.squarespace.com/static/5167f6a2e4b0f1cbdee8d1c0, 520.

xlix. Hill, R. (1998). What Sample Size is 'Enough' in Internet Survey Research? Interpersonal Computing and Technology: An electronic. Journal for the 21st Century, 2 (1), 15-30, http://www.emoderators.com/ipct-j/1998/n3-4/hill.hmtl

l. Hueber, C., Horejsi, K., & Schledjewski, R. (2016). Review of Cost Estimation: Methods and Models for Aerospace Manufacturing. Advanced Manufacturing Polymer and Composite Science (AMPCS), Vol. 2, No. 1, 1-13. ISSN: 2055-0340 (Print) 2055-0359 (Online). https://www.tandfonline.com/loi/yadm20.

li. IEEE Computer Society. Software Engineering Standards Committee, & IEEE-SA Standards Board (1998). IEEE recommended practice for software requirements specifications. Institute of Electrical and Electronics Engineers, 1-40.

lii. International Records Management Trust (1999), "Preserving Records", 5 (2), 12-21, http://www.irmt.org/documents/educ_training/...rec/IRMTpreserve_ recs.doc

liii. Isaias, P., & Issa, T. (2015). Information Systems Development Methodologies in High Level Models and Methodologies for Information Systems. Springer New York, 41-61.

liv. Jarke, M., Lenzerini, M., Vassiliou, Y., & Vassiliadis, P. (2013). Fundamentals of data warehouses. Springer Science & Business Media, 11-20.

lv. Karagiannis, D. (2015, October). Agile modeling method engineering. In Proceedings of the 19th Panhellenic Conference on Informatics (pp. 5-10). ACM.

lvi. Khan, P. M., & Beg, M. S. (2013, April). Extended decision support matrix for selection of sdlc-models on traditional and agile software development projects. In 2013 Third International Conference on Advanced Computing and Communication Technologies (ACCT) (pp. 8-15). IEEE.

lvii. Krejcie, R.V. & Morgan, D.W. (1970). Determining Sample Size for Research Activities. In Hill, R. (1998): "What Sample Size is 'Enough' in Internet Survey Research"? Interpersonal Computing and Technology: An electronic Journal for the 21st Century, 13-25, http://www.emoderators.com/ipct-j/1998/n3-4/hill.hmtl.

lviii. Kroeneke, K., & Hattach F. (2000). SCRUM: A Pattern Language for Hyper productive Software Development, Pattern Languages of Program Design. Reading, MA: Addison-Wesley, 4.

lix. Kumar, N., Zadgaonkar, A. S., & Shukla, A. (2013). Evolving a new software development life cycle model SDLC-2013 with client satisfaction. International Journal of Soft Computing and Engineering (IJSCE), 3 (1), 2231-2307.

lx. Labra, J. G. B., Brewster, M., Tang, Z., Jimenez-Salgado, R., Waldman, L. F., Woo, J. W. Y., & Henderson, J. T. (2016). U.S. Patent No. 9, 292, 360. Washington, DC: U.S. Patent and Trademark Office, 20-35.

lxi. Leau, Y. B., Loo, W. K., Tham, W. Y., & Tan, S. F. (2012). Software development life cycle AGILE vs. traditional approaches. In International Conference on Information and Network Technology, 37 (1), 162-167.

lxii. Madukosiri, C. H., & Bawo, D. S. (2012). Illness pattern and the relationship between the Prevalence of malaria and other infections in Niger Delta University. Agriculture and Biology Journal of North America, 2151-7525.

lxiii. Mana, W., & Kanthawongs, P. (2012, June). Perceived Ease Of Use, Perceived Usefulness, and Self-Efficacy toward Course Quality of Hybrid Instruction Model. In IABE-2012 Venice-Summer Conference, 335.

lxiv. Mangin, J. P. L., Bourgault, N., Guerrero, M. M., & Egea, J. M. (2011). Modeling perceived usefulness on adopting on line banking through the tam model in a Canadian banking environment. Journal of Internet Banking and Commerce, 16 (1), 13.

lxv. Masila, C. K. (2014). Integrating Item Response Theory of testing for Quality Tests by Implementing an Online Item Development and Banking System for the Kenya National Examinations Council. Doctoral dissertation, University of Nairobi, 35-55.

lxvi. Maxwell, J. A. (2012). Qualitative research design: An interactive approach: An interactive approach. Sage Publication, 22-43.

lxvii. McKemmish, S. (2001). Placing records continuum theory and practice. Archival science, 1(4), 333-359.

lxviii. Michaels, D. (2013). Scrum Agile framework for completing complex projects, Version 0.9. By Tutorials Point, 1-41, http://tutorialspoint.com.

lxix. Mills, J. A., Teplitsky, C., Arroyo, B., Charmantier, A., Becker, P. H., Birkhead, T. R., & Bushuev, A. (2015). Archiving primary data: solutions for long-term studies, Trends in ecology & evolution, 30 (10), 581-589.

lxx. Murray, J. (2013). Likert data: What to use, parametric or non-parametric? International Journal of Business and Social Science, 4 (11), 13-14.

lxxi. Musil, J., Musil, A., & Biffl, S. (2013, August). Elements of software ecosystem early-stage design for collective intelligence systems. In Proceedings of the 2013 International Workshop on Ecosystem Architectures, ACM, 21-25.

lxxii. Nelson, M. R. (2009, October). Briefing Paper On Cloud Computing And Public Policy, Prepared For The Iccp Technology Foresight Forum1. DSTI/ICCP (2009), 172.

lxxiii. Ngoepe, M. S. (2008). An exploration of records management trends in the South African public sector: A case study of the department of provincial and local government. M.Sc. Thesis submitted to Information Science department at the University of South Africa, 29-54, http://uir.unisa.ac.za/bitstream/handle/10500/2705/dissertation.ngoepe %20m.pdf?

lxxiv. Nkwocha, P. C. (2010). Educational research: Principles and applications. Owerri, Imo State: Kandid Gold Publishers, 40-45.

lxxv. NUC Monday Bulletin (2018, June). A Publication of the Office of the Executive Secretary. 12 (23), 19-20, ISSN: 0795-3089.

lxxvi. Nwaomah, A. E. (2015, June).Information And Communication Technology (ICT) Usage on Students Records Management Effectiveness in the Nigerian Universities. IJournals: Int. Journal of Social Relevance & Concern (IJSRC) ISSN: 2347-9698, 3(6), 15-22.

lxxvii. Obasi N., Nwachukwu E. O., & Ugwu C. (2013, June). A Novel Web-Based Student Academic Records Information System, West African Journal of Industrial and Academic Research, 7 (1), 31-47.

lxxviii. Obiniyi, A. A., & Ezugwu, E., A. (2010). Design and Implementation of Students' Information System for Universities using Neural Networks: An Open Source Approach. International Journal of Green Computing, 1 (1), 1-15.

lxxix. Okebukola, P., (2010, September). Fifty Years of Higher Education in Nigeria: Trends in Quality Assurance. Presentation at the International Conference on the Contributions of Nigerian Universities to the 50th Independence Anniversary of Nigeria, 27-29.

lxxx. Oliveira, T., Thomas, M., & Espadanal, M. (2014). Assessing the determinants of cloud computing adoption: An analysis of the manufacturing and services sectors. Information & Management, 51 (5), 497-510.

lxxxi. Peltz, C. (2003). Applying Design Issues and Patterns in Web Services. Dev. Resource Journal, 15, http://devresource.hp.com/drc/technical_articles/ WSsuccess1.pdf.

lxxxii. Petroshius, S. M. (2015). The Marketing Research Principles Course: An Alternative Approach to the Client-Sponsored Project. In Proceedings of the 1987 Academy of Marketing Science (AMS) Annual Conference, Springer Inter. Publishing, 222-226.

lxxxiii. Philips, S. (2014). The basics of scrum; an introduction to the framework. By Scrum Inc., 2-9, http://scruminc.com.

lxxxiv. Popovič, A., Hackney, R., Coelho, P. S., & Jaklič, J. (2012). Towards business intelligence systems success: Effects of maturity and culture on analytical decision making. Decision Support Systems, 54(1), 729-739.

lxxxv. Powell, S. & Baker, K. (2009). Management Science: The Art of Modeling with Spreadsheets. 2nd edition, John Wiley & Sons, 23-25.

lxxxvi. Power, D. J., Sharda, R., & Burstein, F. (2015). Decision support systems. John Wiley & Sons, Ltd, 31-38.

lxxxvii. Rad, N. K. & Turley, F. (2013). The Scrum Master Training Manual, A Guide to Passing the Professional Scrum Master (PSM) Exam, version 1.2. Published by Management Plaza, iPlazza N.V. Barbarastraat13 bus 1, 3120 Tremelo, Belgium, 10-84, http://mplaza.pm/product/scrum-master-training-manual/

lxxxviii. Raffensperger, J. F. (2001, July). New Guidelines for Spreadsheets. Proceedings of the European Spreadsheet Risks Interest Group, Amsterdam, 61-76. http://arxiv4.library.cornell.edu/ftp/arxiv/papers/0807/0807.3186.pdf.

lxxxix. Reeder, L. (2014). Core Scrum v2014.08.15. By Scrum Alliance, 7401 Church Ranch Blvd. #210, Westminster, CO 80021 USA, 1-13, https://www.scrumalliance.org/why-scrum /core-scrum-values-roles

xc. Reeder, L. (2015). What is Scrum Backgrounder? By Scrum Alliance, 7401 Church Ranch Blvd. #210, Westminster, CO 80021 USA, 1-2, https://www.scrumalliance.org/scrum/media/ScrumAllianceMedia/Public%20 Relations_2/What-is-Scrum-Backgrounder-2015.pdf

xci. Ritchie, J., Lewis, J., Nicholls, C. M., & Ormston, R. (2013). Qualitative research practice: A guide for social science students and researchers. Sage Publications,12-16.

xcii. Robertson, P. (1997, May). Integrating Legacy Systems with Modern Corporate Applications. Communications of the ACM, 40 (5), 39-46.

xciii. Roy, R. (2003). Cost Engineering: Why, What and How? Decision Engineering Report (DEG) series, Cranfield, UK. https://www.dspace.lib.cranfield.ac.uk/handle/1826/64.

xciv. Rush, C., & Roy, R. (2000). Analysis of Cost Estimation Processes used within a concurrent engineering environment throughout a product life cycle. Proceedings of CE Conference, 58-67.

xcv. Sagiroglu, S., & Sinanc, D. (2013, May). Big data: A review. In 2013 International Conference on Collaboration Technologies and Systems (CTS) (pp. 42-47). IEEE.

xcvi. Schubel, P., J. (2012). Cost Modelling in Polymer Composite Applications: Case study – Analysis of Existing and Automated Manufacturing Processes for a Large Wind Turbine Blade. Composites Part B, 43, (3), 953-960, doi: 10.1016/j.compositesb. 2011.11.036.

xcvii. Schwaber, K., & Sutherland, J. (2014). Scrum Guide. By Scrum.org., 2-21, https://www. scrum.org/Resources/What-is-Scrum

xcviii. Schwaber, K., & Sutherland, J. (2015). The Scrum Guide; The Definitive Guide to Scrum: The Rules of the Game. By Scrum.org and Scrum Inc., 3-16, https://www.scrum.org/ Resources/What-is-Scrum

xcix. Scot, R. (2007, July). Software Project Test Plan, Project: ADS – Ambulance Dispatch System. Software Development Journal, 5 (5), 21-31.

c. Shepherd, E., & Yeo, G. (2011). Managing Records: A Handbook of Principles and strategic resource in the Government Ministries of Tanzania. Information Development, 27 (4), 264-278.

ci. Shin, D. H. (2016). Cross-Platform Users' Experiences Toward Designing Interusable Systems. International Journal of Human-Computer Interaction, 1-12.

cii. Sommerville, I. (2016). IEEE Software and Professional Development. IEEE Software, 33 (2), 90-92.

ciii. Spee, P., Jarzabkowski, P., & Smets, M. (2016). The influence of routine interdependence and skillful accomplishment on the coordination of standardizing and customizing. Organization Science, 27 (3), 759-781.

civ. Spreadsheet Analytics (2010). Checking & Auditing in Spreadsheet Analytics: Resources for Spreadsheet Analysts. 1-25, http://www.usfca.edu/bps/ spreadsheet-analytics/development-management /checking-auditing.

cv. Sun Microsystems (2004). Designing Web Services with the J2EE(TM) 1.4 Platform: JAX-RPC, SOAP, and XML Technologies. Sun Publications, 4 (6), 17-30. http://java.sun.com/blueprints/guidelines/designing_ webservices/html/

cvi. Theunissen, W., Boake, A., & Kourie, D. (2005). In search of the sweet spot: Agile open collaborative corporate software development. Proceedings of the 2005 Annual Research Conference of the South African Institute of Computer Scientists and information Technologists on IT Research in Developing Countries, White River, South Africa, 268-277.

cvii. Ukem, E., O., & Ofoegbu, F., A. (2012). A Software Application for University Students Result Processing. Journal of Theoretical and Applied Information Technology, 35(1), 14-18, ISSN: 1992-8645 (print), E-ISSN: 1817-3195, www.jatit.org.

cviii. Upward, F. (1997). Structuring the records continuum, part two structuration theory and record keeping. Archives and Manuscripts, 25 (1), 1-11.

cix. Upward, F. (2000). Modeling the continuum as paradigm shift in record keeping and archiving processes and beyond – a personal reflection. Records Management Journal, 10 (3), 115-139.

cx. Ward, J., & Peppard, J. (2016). The Strategic Management of Information Systems: Building a Digital Strategy. John Wiley & Sons, 18-23.

cxi. Xu, X. (2012). From cloud computing to cloud manufacturing. Robotics and computer-integrated manufacturing, 28 (1), 75-86.

cxii. Yin, R. K. (2013). Case study research: Design and methods. Sage publications, 23-35.

cxiii. Youh, D. X. (2010, November). Design and Implementation of a Client Server Distributed Database for Student Result Processing. The Pacific Journal of Science and Technology, 11 (2), 288-295, http://www.akamaiuniversity.us/PJST.htm.

cxiv. Zhang, Y., Wang, S., Dong, Z., Phillip, P., Ji, G., & Yang, J. (2015). Pathological brain detection in magnetic resonance imaging scanning by wavelet entropy and hybridization of biogeography-based optimization and particle swarm optimization. Prog Electromagn Res, 152, 41-58.