MADONNA
UNIVERSITY

# INTERNATIONAL JOURNAL OF EDUCATION AND ARTS

VOL. 1 , NO 4

2023

International Journal of Education and Arts Vol.1 , N0 4 Nov. 2023

# Analysis of Solution Domain skills acquisition in Software Engineering in Computer Technology among Computer Education students in Universities in South-East Nigeria

**Dr. Ibebuife Enemchukwu UGWU & Rolland Offorbike UGWOKE**
Department of Computer Science Education
Faculty of Education and Arts
Madonna University Nigeria
Email: ibebuife.ugwu.pg41229@unn.edu.ng

## Abstract

*The study surveyed the Solution Domain skills acquisition in Software Engineering in Computer Technology by Computer Education students of Universities in South-East Nigeria. Specifically, the study sought to analyze the software design skills in Software Engineering needed by Computer Education students of Universities, software implementation skills in Software Engineering needed by Computer Education students of Universities and software maintenance skills in Software Engineering needed by Computer Education students in Universities. The study sought answers what are the software design skills in Software Engineering needed by Computer Education students in Universities, the software implementation skills in Software Engineering needed by Computer Education students in Universities and the software maintenance skills in Software Engineering needed by Computer Education students in Universities. The* hypotheses formulated include that *there is no significant difference in the mean ratings of opinions of Lecturers and Technologists on the software design skills in Software*

*Engineering needed by Computer Education students, there is no significant difference in the mean ratings of opinions of Lecturers and Technologists on the software implementation skills in Software Engineering needed by Computer Education students, and that there is no significant difference in the mean ratings of opinions of Lecturers and Technologists on the software maintenance skills in Software Engineering needed by Computer Education students. The study adopted the survey research design with a population of 104 respondents comprised 61 Lecturers and 43 Technologies in Computer Education. Mean and standard deviation were used to answer the research questions while the correlated t-test statistic was used to test the null hypotheses at 0.05 level of significance. The study reveals that considerations of compatibility, maintainability, reliability, reusability and security are key design skills in Software Engineering. The study also reveals that testing skills that include item testing, unit testing and entire program (system) testing are highly needed in software engineering. The study finally revealed that the maintenance skills required include that of Creation of maintenance plan, Follow-up on product configuration, Implementation of the modification and Consideration of software retirement. In conclusion, there are needs for the acquisition of software design skills, software implementation skills and software maintenance skills in software engineering. It is recommended that university students should focus on solutions domain skills as a major emphasis in software development.*

**Keywords:** Analysis of Solution, Domain, skills acquisition, Software Engineering and Computer Technology

## Itroduction

Knowledge is acquired by undergoing through a series of learning experiences. Skill acquisition involved the gaining of psycho-motor skills essential for making a beginning and advancing in occupations related to one's area of interest. Skill is defined as the learned ability to bring about an end result or objective with maximum certainty and efficiency but with minimum time or energy or both (Savitch, 2017). Software is defined as the whole set of programs, procedures, instructions and routines associated with the operation of a Computer. Software engineering is defined by the US Bureau of Labour Statistics (BLS) in Jegede and Owelabi (2019) as the use of the knowledge of science and mathematics to design, build and test computer and computer software.

Skill acquisition in Software Engineering can be perceived as a process which involves teaching and learning including classroom instruction and out of classroom preparation for instruction. A learner is said to have acquired the needed skills when the stated objectives are achieved, that is when the expected change in behavior or skill is observed in the learner.

The acquisition of well-developed software engineering skills is through well-planned practical oriented lessons. Savitch (2017) citing Computing Sciences Accreditation Board (2016) stated that practical work is seen as providing a way for developing a number of different important practical skills, because it provides interest and enjoyment; produces enthusiasm, encourages initiative, imagination and cooperation and develops self-reliance. Software engineering is an aspect of the newly introduced information and communication technology.

The early Nigerian Educational system being a hand down of the colonial power was void of Technology. Then the federal government of Nigeria had a review of the educational system to make adequate provision for practical skill acquisition in Technology. According to Chukwuemema and Anaele (2015) the Nigerian Government having foreseen the future role of technology changed and expanded her educational system to meet the present and future world characterized by the technological and scientific trends of development. Hence Introductory Technology was introduced and made compulsory in the Nigerian Junior secondary schools (FRN, 1998).

Not long ago, there was an advance in technology that brought about the new field 'Information Technology' sometimes referred to as Computer Technology. Computer Technology comprises some major Areas/branches which include among others software engineering, graphics & multimedia, network administration and security, telecommunications, hardware management etc. The acquisition and development of skills in these branches of Computer Technology especially in software engineering is essential in our present society. Yang, Hicks and Chang (2015) supported Dasuki and Quaye (2016) identified that lack of skill acquisition in Software engineering is as a result of poor teaching and learning. Consequently, this problem of difficulty in skill acquisition resulting from poor teaching and learning in computing necessitated the reason why Computer Education programme was introduced (Nigeria National Computer Policy, 1995).

Computer education is a programme designed to educate a learner on Computers and their operations (that includes Software engineering) as well as how to educate others with the knowledge acquired. Computer education according to Jegede and Owelabi (2019) is an education which equips the learners with the overall knowledge of computer and computing as well as the pedagogical skills and techniques in imparting the same knowledge to others. The Computer education curriculum in Nigerian Universities has clearly indicated the courses required to be mastered by the students in the program. Some of the main courses in Computer education that must be mastered by all the students in the program is Software Engineering (Programming) or Software Development (CSAB, 2016). Hence, a student having undergone through Software engineering training in Computer education, is not only expected to have acquired knowledge and skills in Software engineering for IT industrial operations, but also be equipped with pedagogical skills and techniques in Software engineering education and training (Aayushi, 2020). One of the major branches of Computer Technology as outlined above is Software Engineering.

Provisions are made in the software engineering curriculum for possible skills in the ways of analyzing users' requirements, developing system specification, process of software design, and the strategies for software implementation. Other provisions made include the software testing and deployment strategies, and the maintenance process (SWEBOK, 2019). These provisions are broadly categorized into the problem domain and the solution domain. The problem domain includes analysis of users' requirements, and development of system specification while other provision which are software design, implementation and maintenance fall under the solution domain.

Software design of the solution domain is a process to transform user requirements into some suitable form, which helps the programmer in software coding and implementation (Rouse, 2019). Dasuki and Quaye (2016) describing software design stated that Software design is the first step in SDLC (Software Design Life Cycle), which moves the concentration from problem domain to solution domain. The author also described implementation as the coding phase during which time the design is translated into a programming language.  Hicks and Chang (2015) in the same vein described maintenance as an activity of making sure that the software is flexible, scalable and can be improved on. Provisions are made for skill acquisition in these aspects of solution domain of Software Engineering by students (CSAB, 2016).

Thirteen universities offer Software Engineering programme in South-East geopolitical zone of Nigeria (Chima, 2021). However, available records show that students who undertake the courses directly related to skill acquisition in software engineering perform below average. According to Galadanci, Mukhtar and Muaz (2019), the average performance of students in Software engineering especially in developing countries such as Nigeria is below average when compared with those in the developed world.

It appears that the processes of software design, implementation and maintenance are not effective. According to Savitch (2017), the goal of the software design and implementation is to take the user requirements as challenges and tries to find optimum solution. While the software is being conceptualized, a plan is drawn out to find the best possible design for implementation (coding). The author noting two distinctly different approaches available that are the traditional design approach and the object-oriented design approach stated that Object-oriented design approach enables a design that mimics the real world situation.

It also seems as though the implementation (coding) process is ineffective. Rouse (2019) noted that the purpose of the implementation phase (sometimes called the coding phase) of software development is to translate the software design into source code which is the source of the program (SWEBOK, 2019*)*. The author added that program testing both unit, integration and system testing are part of the implementation and require expertise and skill as failure in perfect testing results in logical error.

Maintenance as the final solution domain phase ensures that the software is scalable, flexible and can be improved on. Maintenance of a typical software product requires much more than the effort necessary to develop the product itself (Savitch, 2017). Maintenance is also identified to be categorized into any of these three kinds of activities: Corrective maintenance that involves correcting errors that were not discovered during the product development phase, Perfective maintenance that has to do with improving the implementation of the system, and enhancing the functionalities of the system according to the customer's requirements or Adaptive maintenance which has to do with porting the software to work in a new environment (SWEBOK, 2019).

The outcome of the above system is that Computer students, who read software engineering, graduate without acquiring the skills required to function as a professional software engineer. According to Dasuki and Quaye (2016) many students of software engineering graduate unemployable, in the sense that they graduate without being able to tackle the problems of design, implementation and maintenance of the software product, thereby, being unable to apply their school experience in their job areas. Thus, there is a need for a survey of the solution domain skills needed for acquisition in Software Engineering.

## Statement of the Problem

In Computer Technology fields more especially in Computer Education, software engineering involves a lot of diverse practical skills acquisition categorized into the problem domains skills and solution domains skills. It is expected that when appropriate hardware, software and human resources are put in place, students should be able to identify and learn the needed problem domain skills in software engineering without biases on the requisite skills in the domain. Students therefore ought to learn and be furnished the skills in software design, software implementation and software maintenance.

Unfortunately, it appears Computer Education in Nigeria, students are unable to identify and learn the needed problem domain skills in software engineering. Students seem capable to identify, learn and have the skills in software design, software implementation and software maintenance. This results in most students dropping out of schools owing to biases while majority graduate without a grip of the skills needed for employment in their field.

In view of this, a gap exist between what ought to be and what is and this gap is what this study sought to fill by analyzing the Solution Domain skills required for acquisition in Software Engineering by Computer students of Universities in South-East Nigeria.

## Purpose of Study

The main purpose of the study is to analyze the Solution Domain skills needed in Software Engineering by Computer students. Specifically, the study sought to analyze the

1. software design skills in Software Engineering needed by Computer Education students of Universities
2. software implementation skills in Software Engineering needed by Computer Education students of Universities
3. software maintenance skills in Software Engineering needed by Computer Education students in Universities

## Research Questions

1. What are the software design skills in Software Engineering needed by Computer Education students in Universities?
2. What are the software implementation skills in Software Engineering needed by Computer Education students in Universities
3. What are the software maintenance skills in Software Engineering needed by Computer Education students in Universities

## Hypotheses

$Ho_1$: There is no significant difference in the mean ratings of opinions of Lecturers and Technologists on the software design skills in Software Engineering needed by Computer Education students

$Ho_2$: There is no significant difference in the mean ratings of opinions of Lecturers and Technologists on the software implementation skills in Software Engineering needed by Computer Education students

$Ho_2$: There is no significant difference in the mean ratings of opinions of Lecturers and Technologists on the software maintenance skills in Software Engineering needed by Computer Education students

## Methodology

This study adopted the survey research design as most appropriate. The design method is suitable because it intends to survey the opinion of Lecturers and Technologists currently teaching Software Engineering courses at the university level on the skills in Software Engineering. This study was conducted in the five States namely Abia, Anambra, Ebonyi, Enugu and Imo States in the South-East geopolitical zone of Nigeria. The population for this study is 104 respondents comprising 61 Lecturers and 43 Technologies currently teaching software engineering courses in all the 13 universities in the South-East Geopolitical zone of Nigeria. The entire population of lecturers of software engineering courses was used. The instrument for data collection in this study was a structured questionnaire containing items based on the research questions of the study. The research instrument was subjected to face validation by three experts in the Computer Education Department, University of Nigeria, Nsukka. Reliability of the instrument was determined using Cronbach Alpha formula. The internal consistency of the items was established by a singular administration of the instrument to Madonna University, Nigeria. The reliability coefficient of the sections all together was 0.91. The data collected were analyzed using mean and standard deviation to answer the research questions and the correlated t-test statistics to test the null hypotheses at 0.05 level of significance. In the first and second table analyses, items with a mean rating of 2.50 and above were regarded as Agree while any item with a mean rating less than 2.50 was regarded as Disagree.

## Results

### Research Question 1

What are the software design skills in Software Engineering needed by Computer Education students of Universities?

The table 1 below shows that only one item out of the twelve items with mean rating less than 2.50 were rated as Disagree by the respondents as software design consideration skills in Software Engineering that must be possessed by computer students of Universities. Eleven other items were rated as Agree with mean rating more than 2.50. This signifies that all the skills with mean rating of more than 2.50 are required by the Computer students

**Table 1:** ***Mean and Standard Deviation on responses of Respondents on the software design skills in Software Engineering needed by Computer Education Students of Universities***

| S/N | Design considerations skills in software design | MEAN ($x$) | STDEV (SD) | DECISION |
|---|---|---|---|---|
| 1 | Compatibility - The software is able to operate with other products | 3.43 | 0.19 | Agree |

| | | | | |
|---|---|---|---|---|
| 2 | Extensibility - New capabilities can be added to the software | 3.54 | 0.28 | Agree |
| 3 | Fault-tolerance - The software is resistant to and able to recover from component failure. | 3.14 | 0.64 | Agree |
| 4 | Maintainability - The software can be restored to a specified condition within a specified period of time | 3.10 | 0.66 | Agree |
| 5 | Marketability - If the software is to be mass - marketed, there must be a market for the software | 1.74 | 1.02 | Disagree |
| 6 | Modularity - the resulting software comprises well defined, independent components. | 3.30 | 0.31 | Agree |
| 7 | Packaging - Printed material such should match the style designated for the target market and should enhance usability | 3.49 | 0.20 | Agree |
| 8 | Reliability - The software is able to perform a required function under stated conditions for a specified period of time. | 3.40 | 0.30 | Agree |
| 9 | Reusability - the modular components designed should capture the essence of the functionality expected out of them and no more or less | 2.98 | 1.32 | Agree |
| 10 | Robustness - The software is able to operate under stress or tolerate unpredictable or invalid input | 2.75 | 0.97 | Agree |
| 11 | Security - The software is able to withstand hostile acts and influences | 3.19 | 0.46 | Agree |
| 12 | Usability - The software user interface must be intuitive (and often aesthetically pleasing) to its target user/audience | 3.00 | 0.62 | Agree |

Table 1 shows that all the skills specified for software design were agreed to by the respondents as needed in software engineering except Marketability which recorded disagree. This indicates that software engineers need to have the skills for writing software that are Compatibility, Extensibility, Fault-tolerant and Maintainable.

## Research Question 2

What are the software implementation skills in Software Engineering needed by Computer Education students of Universities?

**Table 2: Mean and Standard Deviation on responses of Respondents on the software implementation skills in Software Engineering needed by Computer Education Students of Universities.**

| S/N | **Programming language considerations skills** | MEAN ($\bar{x}$) | STDEV (SD) | DECISION |
|---|---|---|---|---|
| 1 | Consideration of OS platform (Windows, UNIX, LINUX) in programming language choice. | 3.49 | 0.22 | Agree |
| 2 | Selection of Programming language based on use (whether education, business, scripting, web etc). | 3.40 | 0.21 | Agree |
| 3 | Selection of Programming language based on age | 2.77 | 0.83 | Agree |
| 4 | Selection of programming language based on wide usage. | 2.78 | 0.87 | Agree |
| | **Testing and debugging considerations skills** | | | |
| 5 | Separate testing of items in FSD | 3.36 | 0.22 | Agree |
| 6 | Unit testing of items in FSD | 3.39 | 0.32 | Agree |
| 7 | Testing of the entire program (en block) | 2.90 | 0.53 | Agree |
| 8 | Comparison of Manual results of program with test run program results | 3.45 | 0.16 | Agree |
| 9 | Testers being cautious of run time and logic errors | 3.29 | 0.35 | Agree |
| 10 | Constant usage of debuggers by Programmers and Testers | 3.09 | 0.65 | Agree |
| 11 | Identifying c orrectness and reliability as the key object of testing | 3.49 | 1.13 | Agree |
| | **Deployment considerations skills** | | | |
| 12 | Installation and activation | 3.27 | 0.25 | Agree |
| 13 | Deactivation ( shutting down any executing component) | 2.81 | 0.39 | Agree |
| 14 | Adaptation (modification of software system previously installed) | 2.96 | 0.87 | Agree |
| 15 | Updating (replacement of earlier version with newer release) | 3.59 | 0.22 | Agree |

Table 2 reveals that all the items were rated Agree with a mean of more than 2.50. This implies that all the skills outline in the table are required skills in software implementation in Software Engineering needed by computer Education students of Universities.

### Research Question 3

What are the software maintenance skills in Software Engineering needed by Computer Education students of Universities?

**Table 3: Mean and Standard Deviation on responses of Respondents on the software maintenance skills in Software Engineering needed by Computer Education Students of Universities.**

| S/N | Maintenance skill | MEAN ($\bar{x}$) | STDEV(SD) | DECISION |
|---|---|---|---|---|
| 1 | Conception and Creation of maintenance plan | 3.45 | 0.23 | Agree |
| 2 | Prior preparation for handling problems identified during development. | 3.47 | 0.26 | Agree |
| 3 | Follow-up on product configuration management | 2.91 | 0.79 | Agree |
| 4 | Maintenance programmer getting and confirming users maintenance requests | 2.61 | 0.78 | Agree |
| 5 | Maintenance programmer obtain ing authorizations to apply modification. | 2.78 | 0.95 | Agree |
| 6 | Consideration of request documentation and solution proposal. | 3.44 | 0.31 | Agree |
| 7 | Implementation of the modification itself. | 3.19 | 0.67 | Agree |
| 8 | Checking modification result with submitted user request. | 3.13 | 0.63 | Agree |
| 9 | Consideration of the retirement of the software. | 2.83 | 1.26 | Agree |

Table 3 reveals that all the items were rated Agree with a mean of more than 2.50. This implies that all the skills outline in the table are required skills in software maintenance in Software Engineering needed by computer Education students of Universities

## Test of Hypotheses

### Hypothesis 1

There is no significant difference in the mean ratings of opinions of Lecturers and Technologists on the software design skills in Software Engineering needed by Computer Education students.

**Table 4: The T-test Analysis of the Responses of Respondents on the software design skills in Software Engineering needed by Computer Education Students**

| S/ No | Design considerations skills in software design | Lecturers $N_1$=61 | | Instructors $N_2$=41 | | t-cal | Test of Signi-ficance (2-tailed) |
|---|---|---|---|---|---|---|---|
| | | $x_1$ | $S_1^2$ | $x_2$ | $S_2^2$ | | |
| 1 | Compatibility - The software is able to operate with other products | 3.46 | 0.17 | 3.20 | 3.39 | 0.50 | NS |
| 2 | Extensibility - New capabilities can be added to the software | 3.53 | 0.27 | 3.54 | 3.55 | -0.02 | NS |
| 3 | Fault-tolerance - The software is resistant to and able to recover from component failure. | 3.07 | 0.64 | 3.61 | 3.21 | -1.09 | NS |
| 4 | Maintainability - The software can be restored to a specified condition within a specified period of time | 3.12 | 0.70 | 3.90 | 3.08 | -1.63 | NS |
| 5 | Marketability - If the software is to be mass -marketed, there must be a market for the software | 1.64 | 1.18 | 2.50 | 1.84 | -2.70 | S |
| 6 | Modularity - the resulting software comprises well defined, independent components. | 3.22 | 0.25 | 3.44 | 3.38 | -0.43 | NS |
| 7 | Packaging - Printed material such should match the style designated for the target market and should enhance usability | 3.51 | 0.18 | 3.56 | 3.47 | -0.09 | NS |
| 8 | Reliability - The software is able to perform a required function under stated conditions for a specified period of time. | 3.43 | 0.33 | 3.50 | 3.37 | -0.14 | NS |

KEY: S = Significant at 0.05 level of significance (α = 0.05)
NS = Not Significant at 0.05 level of Significance

From Table 4 above, the calculated values for Design considerations skills: Marketability, Reusability and Robustness being -2.70, 4.95 and 3.64 respectively are each greater than the table value, the null hypothesis is rejected. However, the calculated values for Compatibility, Extensibility, Fault-tolerance, Maintainability, Modularity, Packaging, Reliability, Security and Usability are each less than the table values, hence we do not reject the null hypothesis.

**Hypothesis 2**

There is no significant difference in the mean ratings of Lecturers and Technologists in the Universities on ways of improving the methods of developing system specification in the acquisition of Software Engineering skills by Computer Education students.

**Table 5: The T-test Analysis of the Responses of Respondents on the software implementation skills in Software Engineering needed by Computer Education Students**

| S/ No | Items | Lecturers $N_1$=61 | | Instructors $N_2$=41 | | t-cal | Test of Significance (2-tailed) |
|---|---|---|---|---|---|---|---|
| | | $x_1$ | $S_1^2$ | $x_2$ | $S_2^2$ | | |
| 1 | Consideration of OS platform (Windows, UNIX, LINUX) in programming language choice. | 3.38 | 0.18 | 3.30 | 3.60 | 0.15 | NS |
| 2 | Selection of Programming language based on use (whether education, business, scripting, web etc). | 3.42 | 0.21 | 3.21 | 3.38 | 0.41 | NS |
| 3 | Selection of Programming language based on age | 2.67 | 0.81 | 1.52 | 2.87 | 2.56 | S |
| 4 | Selection of programming language based on wide usage. | 2.82 | 0.93 | 2.80 | 2.74 | 0.05 | NS |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 5 | **Testing and debuggi ng considerations skills** | 3.27 | 0.15 | 3.12 | 3.45 | 0.28 | NS |
| 6 | Separate testing of items in FSD | 3.41 | 0.33 | 3.68 | 3.37 | -0.52 | NS |
| 7 | Unit testing of items in FSD | 2.95 | 0.65 | 3.25 | 2.84 | -0.68 | NS |
| 8 | Testing of the entire program (en block) | 3.54 | 0.13 | 3.43 | 3.36 | 0.21 | NS |
| 9 | Comparison of Manual results of program with test run program results | 3.34 | 0.39 | 2.91 | 3.24 | 0.87 | NS |
| 10 | Testers being cautious of run time and logic errors | 3.04 | 0.67 | 3.45 | 3.14 | -0.84 | NS |
| 11 | Constant usage of debuggers by Programmers and Testers | 3.58 | 1.10 | 5.43 | 3.40 | -3.44 | **S** |
| 12 | Identifying correc tness and reliability as the key object of testing | 3.21 | 0.21 | 3.40 | 3.34 | -0.37 | NS |
| 13 | **Deployment considerations skills** | 2.92 | 0.26 | 3.12 | 2.70 | -0.48 | NS |
| 14 | Installation and activation | 2.89 | 1.02 | 3.22 | 3.03 | -0.69 | NS |
| 15 | Deactivation ( shutting down any executing component) | 3.67 | 0.16 | 3.52 | 3.51 | 0.28 | NS |

From the analysis on Table 5 above, the calculated values for Consideration of the age of the programming language and Correctness and reliability being 2.56 and -3.44 respectively are each greater than the table value, the null hypothesis is rejected. However, the calculated values for Consideration of OS platform, Programming language choice based on use, testing items on FSD, testing program modules separately and en block, Programmers using debuggers, Installation and activation, Deactivation, Adaptation and Updating are each less than the table values, hence we do not reject the null hypothesis.

**Table 6: The T-test Analysis of the Responses of Respondents on the software maintenance skills in Software Engineering needed by Computer Education Students**

| S/ No | Items | Lecturers $N_1$=61 | | Instructors $N_2$=41 | | t-cal | Test of Significance (2-tailed) |
|---|---|---|---|---|---|---|---|
| | | $x_1$ | $S_1^2$ | $x_2$ | $S_2^2$ | | |
| 1 | Conception and Creation of maintenance plan | 3.45 | 0.14 | 3.45 | 3.45 | 0.00 | NS |
| 2 | Prior preparation for handling problems identified during development. | 3.50 | 0.30 | 3.11 | 3.44 | 0.74 | NS |
| 3 | Follow-up on product configuration management | 2.93 | 0.85 | 3.65 | 2.89 | -1.59 | NS |
| 4 | Maintenance programmer getting and confirm ing users maintenance requests | 2.56 | 0.85 | 1.58 | 2.66 | 2.33 | **S** |
| 5 | Maintenance programmer obtaining authorizations to apply modification. | 2.78 | 1.01 | 1.12 | 2.78 | 3.75 | S |
| 6 | Consideration of request documentation and solution proposal. | 3.48 | 0.33 | 3.68 | 3.40 | -0.38 | NS |
| 7 | Implementation of the modification itself. | 3.30 | 0.77 | 2.80 | 3.07 | 1.05 | NS |
| 8 | Checking modification result with submitted user request. | 3.21 | 0.56 | 3.46 | 3.05 | -0.53 | NS |
| 9 | Consideration of the retirement of the software. | 2.83 | 1.32 | 0.52 | 2.83 | 4.98 | NS |

The t-test analysis in Table 6 reveals that 'Maintenance programmer getting and confirming maintenance' requests from users and 'obtaining authorizations to apply modification' with calculated values 2.33 and 3.75 exceeded the table values and so the null hypothesis is rejected. Other items which include Conception and Creation of maintenance plan, Prior preparation for handling problems, Follow-up, Implementation, Checking modification result and Consideration of the retirement of the software have the calculated values less than the table value and so the null hypothesis is not rejected.

## Discussion

The three Research Questions were answered using mean and standard deviation while the null hypotheses were tested using t-test statistics as evident in Tables 1 to 6. Indications in the statistical analysis of the software design skills in Software Engineering needed by Computer Education students in Universities, show that software design skills in Software Engineering that must be possessed by computer students include considerations of compatibility, extensibility, maintainability, modularity, packaging and reliability of the software. Others include that of reusability, robustness, security and usability. This is in line with Dasuki and Quaye (2016) who identified those skill as required in software Engineering for effective software design. However, marketability is identified not be a design skill needed for design (Savitch, 2017).

In the software implementation skills in Software Engineering needed by Computer Education students in Universities, it was indicated that the language considerations skills include those of Operating System platform, language age and level of wide usage. On Testing and debugging, skills in items testing, unit testing and entire program (system) testing are highly needed in software engineering program, then on Deployment considerations skills, Installation & activation together with Adaptation and Updating are necessary sub-skills needed for software engineering. This is supported by Galadanci and others (2019) who identified that in software implementation skills in Software Engineering, language consideration skills in identifying platform, age and usage is essential, unit and entire program testing skills, and skills in installation, activation and updating are necessary and therefore needed.

In the software maintenance skills in Software Engineering needed by Computer Education students in Universities, the maintenance skills required include that of Creation of maintenance plan, Follow-up on product configuration, Implementation of the modification and Consideration of software retirement. This is in accordance with Savitch (2017) who identified as necessary, the software engineering maintenance skills of creating maintenance plan, implementing the modification in due time and considering the retirement of software when the prospect of updating is not cost effective.

**References**:

Aayushi, J. (2020). Object Oriented Programming – Java OOPs Concepts with Examples. Retrieved from https://www.edureka.co/blog/author/aayushiedureka-co/ on June 19, 2020

Chima, U. (2021).*The Effect of Flipped Learning on Students' Academic Achievement and Interest in Java Programming Language in Federal Colleges of Education in South-East, Nigeria*. Unpublished PhD Thesis, Department of Computer and Robotics Education, University of Nigeria, Nsukka.

Chukwuemema, V. C. & Anaele, E. O. (2015). Information and Communication Technology (ICT) skills for integration into Technical Vocational Education and Training (TVET) for sustainable self-employment among Nigerian youths. Computer Education Research Journal (CERJ) Vol. 2: Issue No. 1

Computing Sciences Accreditation Board (CSAB, 2016). CSAB Leading Computer Education".Retrieved November 19, 2016 from http://www.csab.org/

Dasuki, S., & Quaye, A. (2016). Failure in programming courses in institutions of higher education in developing countries: A Nigerian perspective. The Electronic Journal of Information Systems in Developing Countries, 76(1), 1-18.

Galadanci, B. S., Mukhtar, M. I., & Muaz, S. A. (2019). Correlates of Success in Java Programming: Predicting the Performance of Students in a Java Course from the Results of Lower Level Courses. *Journal of Theoretical and Applied Information Technology*, *97*(6).

Jegede, P. O & Owelabi, J.A.(2019). Computer Education in Nigeria

Schools: Gaps Between Policy and Practice Meridian. A Middle School Technology Journal, 6 (2).

Nick, S. (2019, November 3). *Learn Java 12 Programming*: A *step-*by-*step guide* to *learning essential concepts* in *Java SE 10*, *11*, and *12.* https://www.amazon.com/Learn-Java-Programming-step-step-ebook/dp/

NPE (2004). Federal Republic of Nigeria, National Policy in Education 4th Lagos: Nigerian Education Research and Development Council, 2004.

Rouse, M. (2019 March 24). System software. March 09, 2019 from https://whatis.techtarget.com/definition/system-software

Sarah, L. (2016). Definition object-oriented programming (OOP). Retrieved May 1, 2016 from https://www.techtarget.com/contributor/ Sarah-Lewis

Savitch, W. (2017). *Java: An Introduction to Problem Solving and Programming Plus MyProgrammingLab.* https://dl.acm.org/citation.cfm?id=3161. Retrieved in May 15, 2019

*SWEBOK (2019). Software Engineering Body of Knowledge (Version 3). Retrieved 24 May 2019 from www.swebok.org. IEEE Computer Society.*

Yang, J., Lee, Y., Hicks, D. & Chang, K. H. (2015). Enhancing object-oriented programming education using static and dynamic visualization. In *2015 IEEE Frontiers in Education Conference (FIE)* (pp. 1-5). IEEE.