Digital Innovations & Contemporary Research in SCIENCE, ENGINEERING & TECHNOLOGY
Vol. 5 No. 4. December, 2017

Article Citation Format
B. A Muhammad & I.T Ahmad (2017): A Survey of Skills and Practices Among Software Professionals in the Nigerian Tech Industry. Journal of Digital Innovations & Contemp Res. In Sc., Eng & Tech. Vol. 5, No. 4. Pp 99-110

Article Progress Time Stamps
Article Type: Research Article
Manuscript Received: 7th December, 2017
Review Type: Blind - Expedited
Final Acceptance:: 31st December, 2017
DOI Prefix: 10.22624

A Survey of Skills and Practices Among Software Professionals in the Nigerian Tech Industry

B. A Muhammad & I.T Ahmad
Computer Science Unit
Department of Mathematics
Usmanu Danfodiyo University
Sokoto, Nigeria
aliyu.mbello@udusok.edu.ng, idris.ahmad@udusok.edu.ng

ABSTRACT

The tech industry in Nigeria is struggling to strive and compete with other tech industries. Just like other developing nations, lack of skilled software engineers and other technical workers has been a major factor holding the Nigeria tech industry back. It is, therefore, important to have a critical look at the skills and practices that are currently running the sector in order to identify areas that require improvement. This study, a survey, was conducted on the skills and practices of the professionals in the industry consisting of system analysts, developers, maintainers, lecturers, instructors and other practitioners. In total, 120 IT professionals within the Nigerian IT industry were surveyed. The main instrument of the data collection was an online questionnaire using the survey monkey Platform. The result shows that most of the developers (60%) in the industry do not have much experience in the job, with the majority having just 0 – 5 years in the job. However, more than 40% of the practitioners have the qualification of Bachelor of Science (B Sc.) or more. Also, the dominant programming languages in the Nigeria tech industry are Java and PHP. Many do not know or use edge cutting practices such software design patterns in the job. Some cited lack of enforcement by their organization as the justification for such.

Keywords: Design patterns, software design, IT professionals, programming languages, PHP, Java, survey,

1. INTRODUCTION

Software systems have now taken over every bit of human lives. From education, health, businesses, social interaction to military etc., software systems play vital role in computation, semi-automation and full automation. However, the development of such systems involves tremendous effort and hard work as the process is iterative, error prone and many things can go wrong at any point in time (Fiadeiro 2003). The skills needed for software development ranges from the knowledge or mastery of the programming language(s), experience and using established practices for the development.

Many concepts have been developed to ease or guide the processing of knowledge in software development, including information hiding, modularity, objects, functions and procedures, patterns, and more (Robillard 1999; Smith 2014). The Object oriented programming (OOP) paradigm has dominated the software industry because it leads to flexibility through modular design (Smith 2014). This does not mean that the problems in software development can be eliminated by the OOP alone. This is because the OOP paradigm does not make a programmer more advanced coder but makes him/her more strategic (Smith, 2014). It also makes the code easy to maintain (Lengstorf and Wald 2016). Productive practices such as the design patterns simplifies the design and implementation of the software.

Using the software patterns does not require any knowledge of programming tricks or a particular programming language, but requires the developer to carefully identify the particular instances of a problem and match the approach pattern that addresses that problem. (Devedzic 2002). Each particular design problems encountered in the development process is documented along with the detailed approach used to solve them. This way, the approach could be used many times over without doing it the same way twice (Zhang and Budgen 2013). Each of the successfully implemented approach that addresses a particular design issue is named a pattern that described such problem.

Therefore, a productive coder must embrace such patterns to avoid re-inventing the wheel. This will also produce more elegant and more effective design solutions. It will save the time of the development as time-to-market is always a critical factor for software systems. When communicating to team members, it is always easier, faster and more effective if the established template (pattern) is followed (Coplien 2003; Devedzic 2002). Other factors that improve the tech industry is innovation or adoption of a new technology.

The Nigerian tech industry has veered off a couple of years ago but with little progress compared to its counterpart in other countries. A technical reason for such poor performance of the industry is the quality of the software engineers and other technical workers (Timm 2017). The poor state of the infrastructure and shortage of good talent are also responsible holding back the industry Otegbede (2017). Notwithstanding, the tech companies in Nigeria have made a significant progress despite the challenges pulling them back. Companies like *interswitch*, *Andel* and the e-commerce website *jumia,* have had huge contribution to Nigeria and the Africa as a whole, providing jobs, taxes and customer satisfaction (Timm 2017).

In light of the above scenario therefore, the quality of the software engineers and other technical workers needs attention. What technologies are available in the industry? Which skills do the people have and what practices empowers the industry. This study conducts a cross examination of the sector focusing on the skills of the technical workers especially in the design and development of the software systems. The survey targets 120 practitioners within the Nigerian tech industry (in both IT organisations and academic institutions) with a view to understanding the skills they possessed and the practices they deploy in carrying out their job. In particular, coding practices such as design patterns and the programming paradigm were the analysed.

The main instrument of data collection was questionnaire. In particular, *survey monkey*, an online questionnaire was administered to the respective target groups. The link to the survey was communicated to the target groups via email, WhatsApp groups, Facebook messenger, and text messages. The data collected was analysed using simple frequencies and percentages.

2. STATEMENT OF THE PROBLEM

As pointed out in the introductory part, the Nigerian tech industry is hit by shortage of good talent. Bunmi Otegbade, the managing director of the tech company *Generation Enterprise,* is of the opinion that shortage of good talent does not only mean that there are no enough number of people in the industry but also lack of better software engineers and technical workers is holding the industry back. In addition, the present work force of experts in the industry do not deliver the needed excellence. If the software engineers can be improved, it will be easy towards improving the tech policies in the country. Therefore, there is need to identify the key areas where the software engineers in Nigeria need to improve, establish the scale of the problem and come up with valuable recommendations that will improve the software engineering profession in Nigeria. This is the contribution or research gap that this study fills.

3. METHODOLOGY

This study uses a descriptive survey, a form of quantitative research methodology. This approach emphasises the objective measurement of a concepts which is followed by the statistical, mathematical or numerical analysis of the data collected through polls, questionnaires or surveys. Since this survey captures data once from the respondents using questionnaires, hence it can be said to be descriptive survey (Labaree 2017). This is used because surveys provides the means to gather facts about a topic from the experience and expertise of the experts' community (Zhang and Budgen 2013). In other words, Descriptive surveys are designed to obtain information about the current status of a phenomenon or to answer questions like where, what, how, why, when, and who (Oyemade, Odiagbe and Buhari 2015).

3.1 Target Population

The study investigates the current state of the Nigerian tech industry. This is means that IT professionals in the various organisations in Nigeria are the target population. These professionals include those practice in companies, private and public academic institutions, self-employed practitioners and students in their final year of (tech related) study. The categories of the targeted professionals include: developers (programmers), system analysts and other technical workers related to software development.

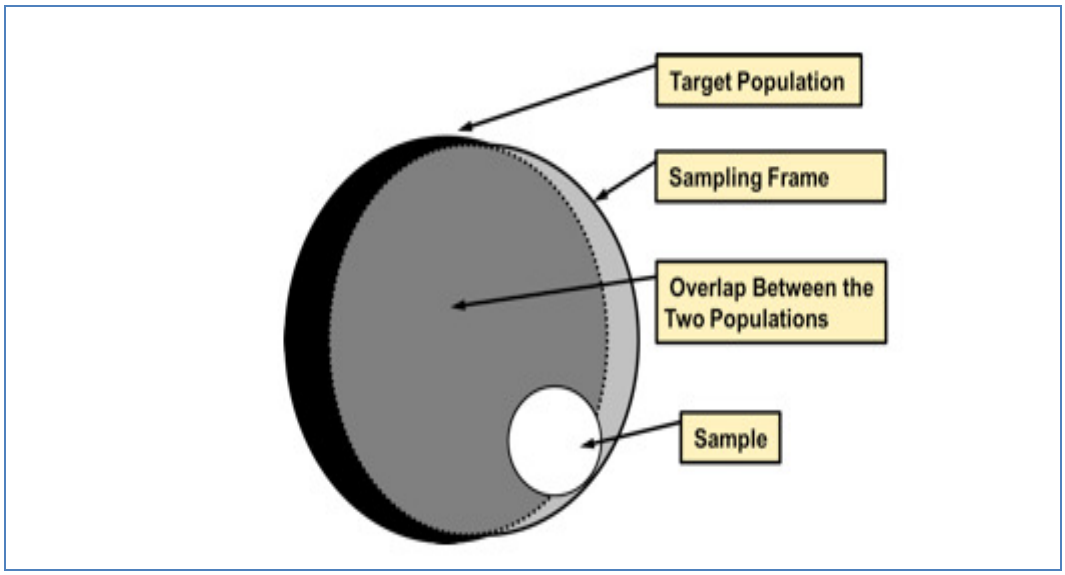

Fig. 3.1 the relationship between the target population and a sampling frame

However, no specifics of the type of software is put into consideration. All software types (web, standalone and mobile) are considered. Knowing or establishing the target population is important in order to obtain the facts that lead to valid conclusions. Also, the target population provides the overall context and represents the collection of people, housing units, schools etc. about which inferences and estimates are desired. A sampling frame represents a list of items in a population from which a sample can be taken. Therefore, the frame will be larger or smaller, depending on practical ways of getting in touch with each member of the sample (Buskirk 2014).

3.2 Sampling Frame

Considering the relatively large size of the tech industry of the country, it is necessary to establish a sampling frame in order to establish representativeness of the sample for generalization. This gives the survey a good coverage of the target population and allows to make inferences that can be put in proper context. This is because, it is difficult to reach every tech professional in Nigeria. There is no specific number of sampling frame but the recommended number which gives the minimum error is used (Science Buddies 2017). The following shows the sample size along with the error associated with it.

Sample Size (*N*)	Margin of Error (fraction)	Margin of Error (percentage)
10	0.316	31.6
20	0.224	22.4
50	0.141	14.1
100	0.100	10.0
200	0.071	7.1
500	0.045	4.5
1000	0.032	3.2
2000	0.022	2.2
5000	0.014	1.4
10000	0.010	1.0

Fig 3.2 the acceptable sampling frame errors Source: (Science Buddies 2017).

As the table shows, using a sample size of 10 means that a margin error of ±0.316 (at 95% confidence interval) i.e. ±31.5% barters the result. This means that the result of 6 out 10 positive is affected by +32% or -32%. Hence, it could be 3 out of 10 or 9 out of 10. As the number of the sample size increases, the lower the margin of error. Therefore, any number from 100 could give a good result with minimum error. A total number of 150 questionnaires were distributed to the target population. Using a stratified random sampling technique, the respondents were selected and sampled.

3.3 Research Instrument

The research instrument used for the survey is the questionnaire. This is because data can be collected relatively quickly because the researcher would not need to be present when the questionnaires were completed. This is useful for large populations when interviews would be impractical. Questionnaires can be administered online without having to physically meet the disparate respondents. An online platform, the survey monkey, was used to generate the questionnaires. The generated link to the questionnaires was sent to the various respondents. However, a problem with questionnaire is that respondents may lie due to social desirability (McLeod 2014). Hence the responses must be validated.

3.4 Validity of the Research Instrument

The questionnaires used for the survey could have bias in them. To validate this instrument is important. The validity was to address bias as the respondents may want to present a positive image of themselves. The instrument was validated by administering the questionnaire a second time to a number of respondents. In most of the cases, the responses from 2 administered questionnaires were exactly the same. This confirms the validity of the responses.

3.5 Response Rate

The approximate acceptable response rate for a questionnaire is 60% (Fincham 2008). However, any response rate above 50% is acceptable (Mugenda and Mugenda 1999). For this survey, out of the 120 questionnaires distributed, 80 were completed and returned. This makes 83.3% return for the study. Hence it is valid for the analysis for go ahead.

4. RESULTS

This section describes the data analysis. The data collected from the respondents was analysed using simple frequencies and percentages. These statistical tools were used because it makes it easier to see what's going on in the data. This means that we could simply count how many responses are obtained each of the questions in the survey that it was possible to obtain. It also computes the percentages from the frequencies easily (Hole 2000).

4.1 Role of the Respondents

The responses show that less than 18% of the practitioners are the actual developers (programmers) in the industry. This is the fraction of those who acquired and use the skills for software development. The majority (33%) end up in the class room of various higher institutions of the country. A good fraction, about 17% (almost the equal number of the developers) have no specific role in the industry. They either provide support for the other roles or are just occupying offices in the various governmental IT departments. The table and the charts below provide the summary of the roles.

Table 4.1 Role of respondents

Role	Response	%
Lecturer	26	32.10
Technician	2	2.47
Developer	16	19.75
System analyst	24	29.63
Others	13	16.05
Total	81	100

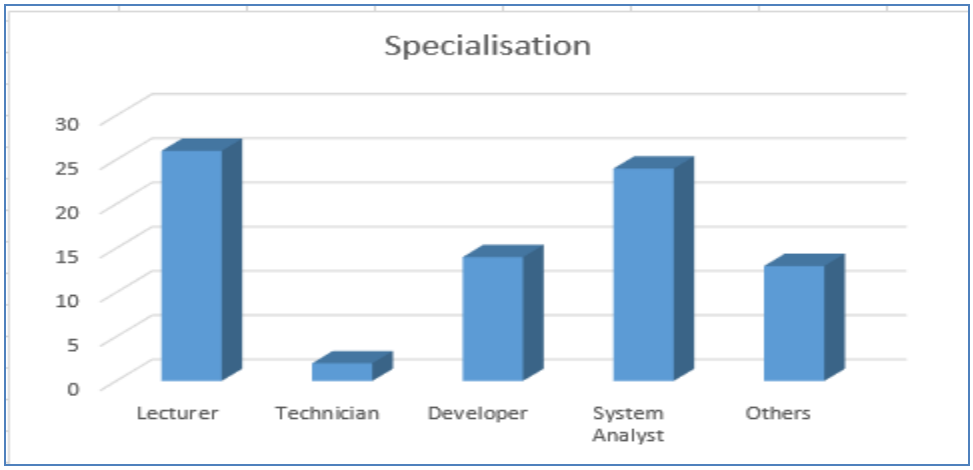

Fig. 4.1 Role of respondents

4.2 Job Experience

From all the developers and other technical workers, majority (about 77%) have little experience in the job. An experience of 5 years or less is very much considered little experience. Real experience begins from 6 years (Siddiqui 2013). None of the respondents (0%) have been in job in the last 15 years. However, the small number of them (about 18%) have a good experience of more than 5 years in the industry. The table 4.2 and fig. 4.2 respectively shows the result of the responses.

Table 4.2 Job experience

Duration	Response	%
0-5	62	77.50
6-10	14	17.50
11-15	4	5.00
>15	0	0.00
Total	80	100

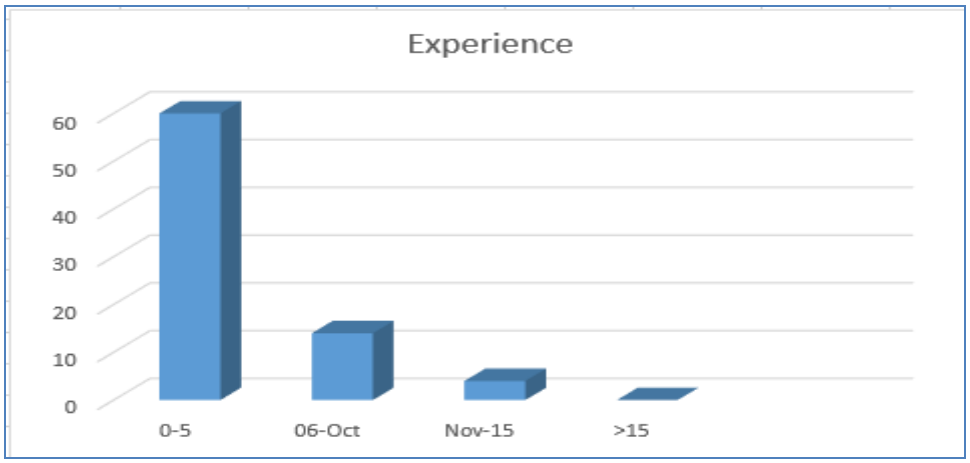

Fig. 4.2 Job experience

4.3 Programming language

Java and PHP are the dominant language of the industry. Together, they made up of more than 60% of the language map of the industry. More than 60% of all the developers are use either Java/PHP or both. However, the percentage of Java is greater than that of PHP because, unlike Java, it supports only web application. Java supports mobile, web and standalone applications development. Therefore, in terms of the web application, PHP is the most popular language among developers. Other languages in the industry include: C, C++, C#, ASP.net, JavaScript etc. Table 4.3 and fig. 4.3 show the results.

Table 4.3 the programming language

Language	Response	%
Java	49	62.03
PHP	45	56.96
C, C++	20	25.32
C#	15	18.99
ASP.Net	14	17.72
Others	26	32.91

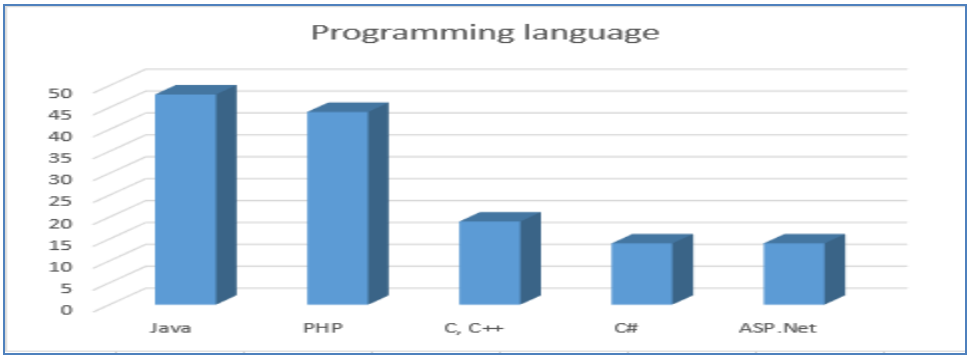

Fig. 4.3 The programming language

4.4 Type of applications

The industry lags behind in mobile applications development. This is indicated by the result as only 15% are involved mobile application development. This has not changed much since the investigation by (Omeruo 2013). This reason for this lack of interest is not far from the marketability of this applications in Nigeria and Africa as a whole. Web applications have dominated the developer's terrain. Again, this can be justified by the need for websites by small, medium and large scale businesses. About 67% of the developers are involved in web application development. Desktop or standalone software practitioners are about 37% of the respondents. About 12% are involved in two (2) or more types of software development. However, there are about 14% who practice nothing in the industry. These are mostly those who occupy positions in government IT departments or top executives of a company who manages the rest of the work force. The table 4.4 and fig. 4.4 below show the response in context.

Table 4.4 Software type

Application	Response	%
Standalone	29	36.25
Mobile	13	16.25
Web	54	67.50
All	10	12.50
None	11	13.75

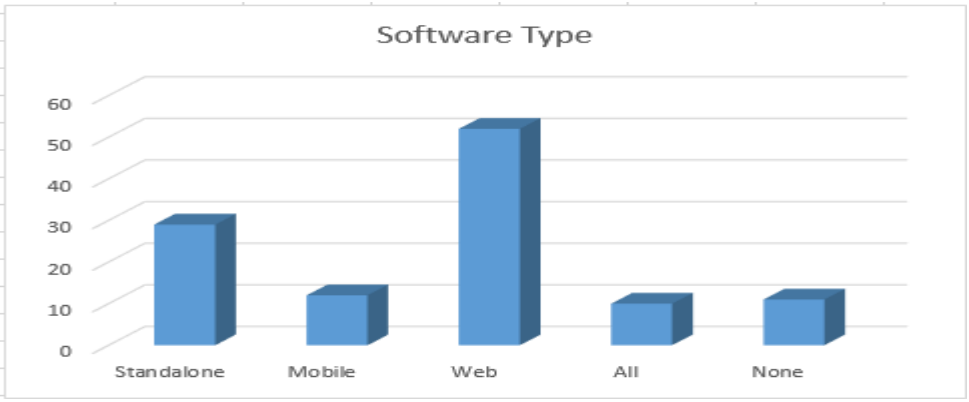

Fig. 4.4 the software type

4.5 Software design patterns

This is one of the excellent programing practice. These techniques ensure efficiency and productivity. Shockingly, nearly more than 30% of the developers have never of the concept of the software design patterns at all. This certainly means that they have never used it at all. One respondent was even confused by the question, not knowing what to say.

Table 4.5 Design patterns

Pattern	Response	%
Yes	52	65.00
No	26	35.00

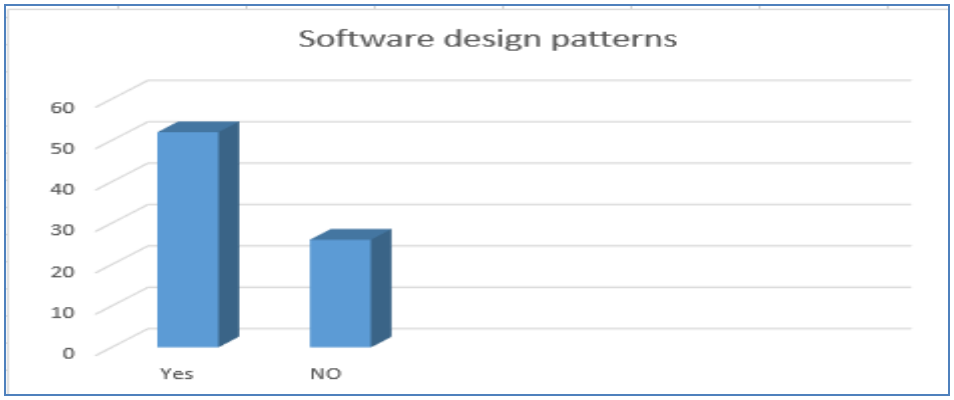

Fig. 4.5 Design patterns

4.6 The Design Patterns

Although there is little acceptance of this patterns ion the Nigerian tech industry, most of the few who accepts them only use the prototype. This is because it is the simplest of the patterns to use. This is shown on the figure and table below.

Table 4.6 Patterns scale

Pattern	Response	%
Singleton pattern	21	30.88
Factory	15	22.06
Builder	12	17.65
Prototype	38	55.88
Observer	6	8.82
Interpreter	19	27.94
Chain of responsibility	7	10.29
Command	17	25.00
Mediator	5	7.35
Adapter	12	17.65
Façade	8	11.76
Bridge	9	13.24
Flyweight	2	2.94
Composite decorator	3	4.48
Proxy	13	19.21

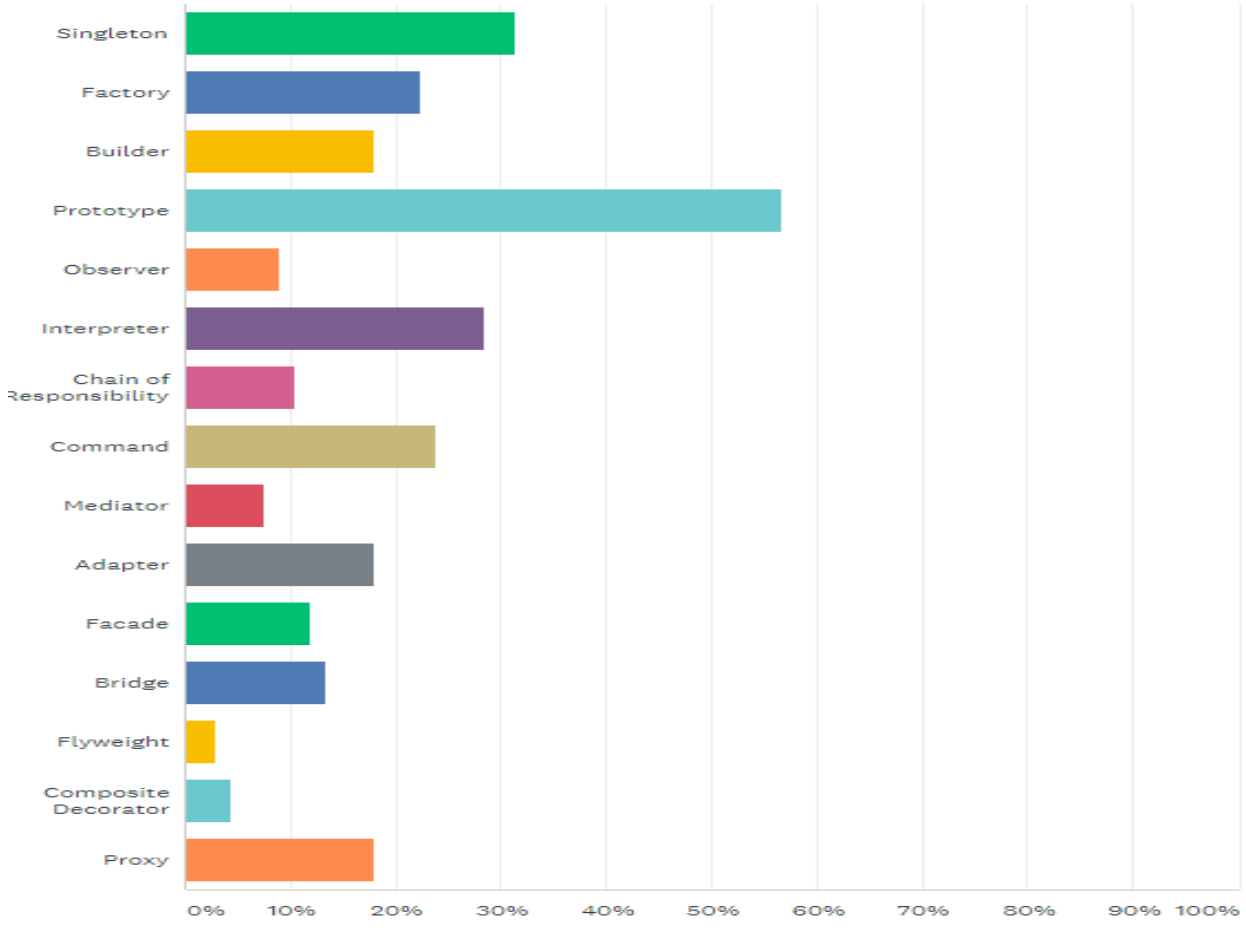

Fig. 4.6 Patterns Scale

The design patterns are used for both design and development. In some cases however, the respondents have no idea about using the patterns. Some responded that they use them for design only, some for implementation and some for both.

4.7 Organisational Enforcement of Patterns

In any organisation, the practices were enforced by the organisation. From the responses however, it is clear that about 45% of respondents' organisations do not enforce any professional practice at all. Another 32% say that their organisation don't mind having the design patterns in development, it is entirely up to developers to deliver their assigned task, the way they do it regardless. Only 23% of the respondents say that their organisations enforce the use of design patterns.

Table 4.7 enforcement of the patterns

Enforcement	Response	%
Yes	17	22.97
No	32	43.24
Don't mind	25	33.78

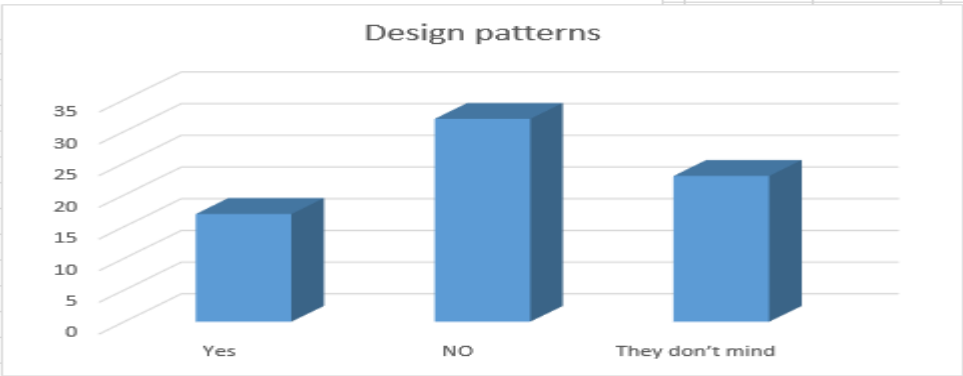

Fig. 4.7 enforcement of the patterns

On the response about their usefulness, majority of the respondents (36.00%) believes that patterns are not useful to the development practice at all, although they do not rule out the fact that they may add value to the development. The following table shows the response.

Table 4.71 usefulness of the patterns

Useful	Response	%
Useful	23	29.87
Not useful	28	36.36
Add no value	0	0.00
Can't say	26	33.77

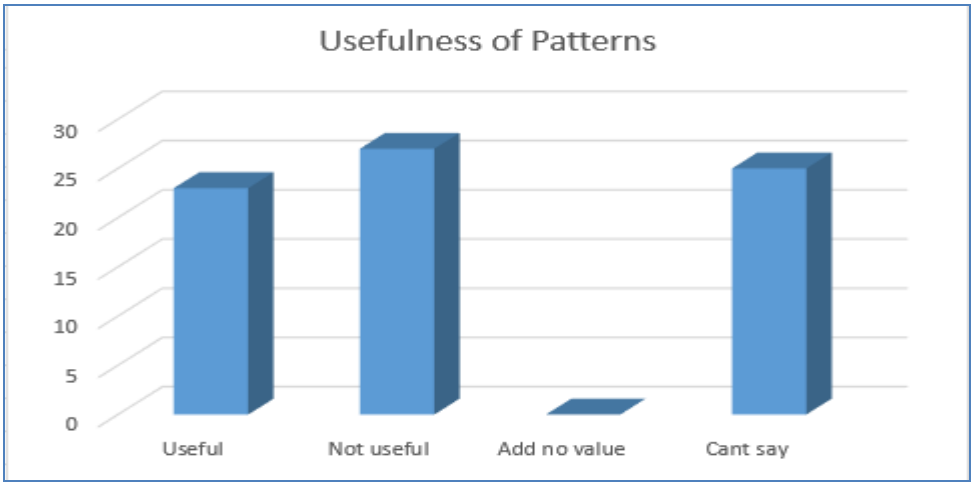

Fig. 4.71 usefulness of the patterns

5. CONCLUSION

From the investigation, it was found that the Nigerian tech industry got workforce with lots of educational qualifications with more than 40% bachelor degree holders and above. However, it has many inexperienced practitioners. Majority of them have 5 or less years of the practice in the job. As the moment, very little can be done to change it. Again, many should be attracted into the job especially by the private companies and start-ups instead of letting them fly abroad. The web applications are the dominate types of application in the industry, partly because most businesses and government departments require a website or portal to run their daily business. Mobile applications and desktop applications mostly have little impact in the industry. Java and PHP have also dominate the programming languages in the industry. This is very clear from the web applications dominance in the sector. Design patterns are import procedures that 'perfect' the software engineering profession. Unfortunately more than 30% of the respondents have never heard of the design patterns at all, let alone use them.

REFERENCES

1. Buskirk, T. (2017) *Target Population and Sampling Frame in Survey Sampling* [online] available from <http://www.theanalysisfactor.com/target-population-sampling-frame/> [September,17 2017]
2. Coplien, J. O. (2003) 'Software Design Patterns'
3. Fincham, J. E. (2008) 'Response Rates and Responsiveness for Surveys, Standards, and the Journal'. *American Journal of Pharmaceutical Education* 72 (2), 43
4. Hole, G. (2000) 'Descriptive Statistics - Frequency Distributions and Averages'. in *Research Methods*. ed. by Anon: COGS, 1-2,3,4,5,6
5. Labaree, R. (2017) *Organizing Your Social Sciences Research Paper: Purpose of Guide* [online] available from <http://libguides.usc.edu/writingguide/methodology> [September/03 2017]
6. Lengstorf, J. and Wald, K. (2016) 'Object-Oriented Programming'. in *Pro PHP and jQuery*. ed. by Anon: Springer, 85-114
7. McLeod, S. (2017) *Questionnaires* [online] available from <https://www.simplypsychology.org/questionnaires.html> [September, 2017 2017]
8. Mugenda, O. M. and Mugenda, A. G. (1999) *Research Methods: Quantitative and Qualitative Approaches.*: Acts press
9. Omeruo, K. (2013) *the Future of Mobile App Development in Nigeria and the Challenge to Nigerian Developers* [online] available from <https://www.techtrendsng.com/future-mobile-app-development-nigeria-challenge-nigerian-developers/> [October/13 2017]
10. Oyemade, O., Odiagbe, J., and Buhari, B. (2015) 'A Survey of Security Vulnerabilities in PHP Applications among IT Professionals in Nigeria'. *African Journal of Computing & ICT* 8 (3)
11. Robillard, P. N. (1999) 'The Role of Knowledge in Software Development'. *Communications of the ACM* 42 (1), 87-92
12. Science, B. (2017) *Sample Size: How Many Survey Participants do I Need?* [online] available from <https://www.sciencebuddies.org/science-fair-projects/references/sample-size-surveys> [Sepetember,17 2017]
13. Siddiqui, F. (2017) *how do You Know when You should no Longer be Classed as a Junior Designer* [online] available from <https://www.bayt.com/en/specialties/q/1134/how-do-you-know-when-you-should-no-longer-be-classed-as-a-junior-designer/> [September/20 2017]
14. Smith, B. (2015) 'Object-Oriented Programming'. in *Advanced ActionScript 3*. ed. by Anon: Springer, 1-23
15. Timm, S. (2017) *Tackle Infrastructure, Skills Challenges to Boost Tech Sector, Say Nigerian Experts* [online] available from <http://ventureburn.com/2017/05/tackle-infrastructure-skills-challenges-boost-tech-sector-say-nigerian-experts/> [September/03 2017]
16. Zhang, C. and Budgen, D. (2013) 'A Survey of Experienced User Perceptions about Software Design Patterns'. *Information and Software Technology* 55 (5), 822-835