

FULOKOJA-2024

FEDERAL UNIVERSITY, LOKOJA, KOGI STATE, NIGERIA-CONFERENCE 2024

The Proceeding of 29th Multidisciplinary Academic Conference, Hummingbird Publications and Research International on African Creativity towards Sustainable Development Goals: Multidisciplinary Approach. Vol. 29 No. 1, 29th-31st January, 2024 at Federal University, Lokoja, Kogi State, Nigeria.

DEVELOPMENT OF AGILE PRODUCTIVITY METRICS OF INDIVIDUAL EXPERT JAVASCRIPT DEVELOPERS FOR SOFTWARE PROJECT MANAGERS

*ABDULGAFAR, A¹, MAKINDE, J,K², AJIBOYE, J.A³, AJIBOYE, M.A⁴

^{1,2}Department of Project Management Technology, Federal University of Technology, Minna, Nigeria. ³Department of Electrical and Electronics Engineering, Federal University of Technology, Minna, Nigeria. ⁴Abuja Electricity Distribution Company (AEDC), ICT Department, Niger Regional Office, Minna, Nigeria.

Abstract

Software Project Managers require metrics to measure productivity in team work. Since agile software development require continuous improvement, metrics helps in identifying bottlenecks and inefficiencies thereby enabling teams to refine their processes iteratively. Productivity metrics also helps in effective resource allocation and optimization to ensure timely delivery of software products by Software Project Managers. Although metrics have been developed for traditional software programmers little work has been done in developing metrics for Agile Software Project Managers specifically for JavaScript Program. In this work, metrics for Individual Expert agile software programmers and specifically for JavaScript was developed. Programs in JavaScript was designed and developed to record the time spent in correcting deliberate errors introduced. Experiment was conducted among one hundred programmers' group of Individual Expert pairs with the aim of recording time spent in debugging the codes. The curve fit regression models of time spent in debugging a number of bugs in agile software written in JavaScript programming language for project managers revealed that Cubic model had the highest R squared value of 0.996 which is closely followed by the quadratic model with a value of 0.980 while the compound, growth and exponential models have the least value of 0.868.

Keywords: Agile, Metrics, Debugging,

Introduction

The art of project management involves keeping track of a team's development to ensure that all project objectives are met within the allocated time frames. Typically, descriptions of these details are included in the project documentation that is created at the beginning of the development process. According to Gemino *et al.* (2021), Manurung and Kurniawan

FULOKOJA-2024

FEDERAL UNIVERSITY, LOKOJA, KOGI STATE, NIGERIA-CONFERENCE 2024

The Proceeding of 29th Multidisciplinary Academic Conference, Hummingbird Publications and Research International on African Creativity towards Sustainable Development Goals: Multidisciplinary Approach. Vol. 29 No. 1, 29th-31st January, 2024 at Federal University, Lokoja, Kogi State, Nigeria.

successfully. The ability of an organization to develop, learn, and advance via a project can be improved with the help of project management knowledge, they are aware of this (Marion *et al.*, 2021; Fixson, 2021).

Software project managers create the project plans for software and online projects and deconstruct the components necessary to finish the project, such as allocating tasks, creating a timeline, and creating checkpoints for progress. Software project managers supervise software projects and the team members involved from beginning to end. Iivari *et al.* (2021).

The Traditional Software Development

Traditional software development makes the assumption that information regarding the clients' demands can be gathered right away Munteanu *et al.* (2021). Traditional software development uses a plan-driven technique and makes the full assumption that all projects needs can be acquired and implemented in advance. The methodology is only appropriate for projects where the technology and requirements are set in stone and where changes are not permitted throughout the entire cycle of software development. There are many different phases in the creation of software that make up the product cycle.

- i. Requirement
- ii. Architecture and Design
- iii. Coding
- iv. Testing and Feedback

The Agile Software Engineering Development

Agile methodologies have outstanding characteristics which are; lightweight, high efficiency, low risk, flexibility, predictability, and scientific method of developing software (Mantovani and Marczak, 2020).

Software Engineers have realized that the traditional, waterfall-style development methods is no longer relevant in the present day fast changing requirements (Lawal *et al.*, 2021; Ogbu, 2021). In traditional software development method, all standard software development phases are followed sequentially and this is known to be the earliest software lifecycle development method which is popular and used in all kinds of projects (Akinsola *et al.*, 2020; Ayegbusi and Adebisi, 2020). More successes are recorded mostly in large and very complex projects. The challenge with the traditional software development method is that all planning is done at the commencement of the software development project. In the waterfall model, requirements cannot be changed during the development stage. This is a flaw in the present-day software development market. A rigid software development method, where changes are difficult will certainly lead to business failure (Núñez *et al.*, 2020).

The waterfall model, shown in Figure 1 therefore, is obsolete. Hence the need for a more flexible model that permits software engineers to change their requirements much later to reflect the changing software business world (Ashmore *et al.*, 2021).

FULOKOJA-2024

FEDERAL UNIVERSITY, LOKOJA, KOGI STATE, NIGERIA-CONFERENCE 2024
The Proceeding of 29th Multidisciplinary Academic Conference, Hummingbird Publications and Research International on African Creativity towards Sustainable Development Goals: Multidisciplinary Approach. Vol. 29 No. 1, 29th-31st January, 2024 at Federal University, Lokoja, Kogi State, Nigeria.

- i. Iterative - this suggests that a software problem can be solved by starting with a small set of elicitation questions and progressively coming up with an approximation of the answer. The design is modified as the specifications are updated to reflect these requirements.
- ii. Incremental - this characteristic demonstrates how the system is broken down into little functional subsystems. The functionality is upgraded together with a new release.
- iii. Self-Organizing - this term refers to a team's autonomy, or the capacity to plan its own structure and tasks.
- iv. Emergent - A technique of working products should arise based on self-organization, and when these forms, a framework of development technologies will also emerge. This is referred to as being emergent.

METHODOLOGY

In order to specify the system requirements intended for the proposed research, this chapter, which focuses on the system methodology, includes topics including research design, participants, procedures and data analysis.

Errors are an inevitable part of producing software, so it is important to establish and implement an effective productivity metrics for Agile project managers on how long it takes a developer to rectify errors. Errors cannot be completely eliminated but it can be undoubtedly lessened and the effects managed. For development of a software that will record time taken to debug errors, some fundamental prerequisites are essential and need to be taken into account. These prerequisites include the following:

- i. Users should be required to sign in before accessing any functions or features of the system, and security access control should be implemented.
- ii. Each member must be fully registered as a user on this platform in order to access the system.
- iii. A user management page should be included in the system so that the system administrator may properly oversee every aspect of the system's operation.
- iv. A listing of registered users on the system.

RESEARCH DESIGN

Visualizing developer productivity in a vacuum is challenging. More broadly, it refers to the quantity of dependable, high-quality software that an engineering team produces quickly. The majority of teams monitor it by evaluating several indicators that show how well a development team works in relation to preset objectives. These consist of teamwork, time-to-completion, bug rate, developer experience and code coverage.

SYSTEM DESIGN

Figure 3 shows the flow chart for developing JavaScript programming languages that is able to measure time spent in error corrections for agile software developers. System design refers to the user interface of the program for debugging errors and the source code.

FULOKOJA-2024

FEDERAL UNIVERSITY, LOKOJA, KOGI STATE, NIGERIA-CONFERENCE 2024

The Proceeding of 29th Multidisciplinary Academic Conference, Hummingbird Publications and Research International on African Creativity towards Sustainable Development Goals: Multidisciplinary Approach. Vol. 29 No. 1, 29th-31st January, 2024 at Federal University, Lokoja, Kogi State, Nigeria.

Table 1: Grouping of Software Programmers (Ajiboye *et al.*, 2016)

Grouping	Remark
Random pairs	Regardless of how long they have been doing pair programming
Expert pairs	Each has over five years of pair programming experience.
Expert-Junior	combination of programmers with one having less expertise in pair programming and the other having more than five years
Junior-Junior	
Individual	Each has fewer than five years of pair programming experience.
Expert	
Individual	more than five years of agile methodology experience
Junior	

Figure 4 shows the JavaScript program interface developed.

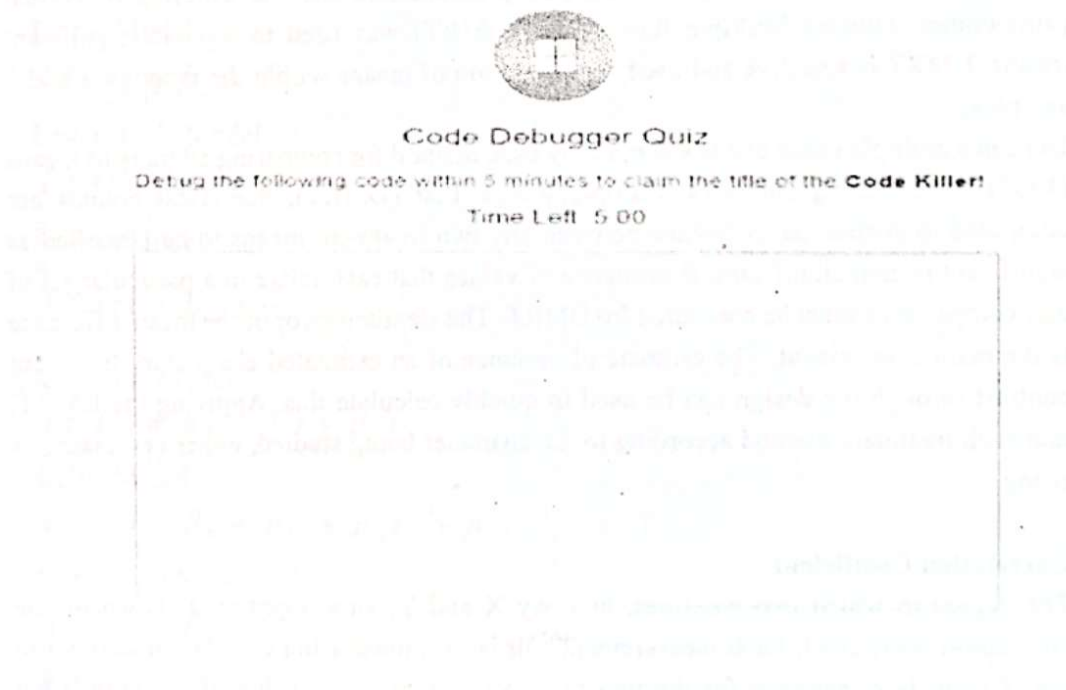


Figure 4: Main Program Interface

DEVELOPMENT OF METRICS

The creation of new productivity measures specifically for software developers is one of the main contributions of this research. These metrics may offer a more objective means of gauging productivity, which could enhance the procedures used in software development and eventually result in better software. Furthermore, the study may advance knowledge of the relationship between personality qualities and training regimens and software development productivity.

FULOKOJA-2024

FEDERAL UNIVERSITY, LOKOJA, KOGI STATE, NIGERIA-CONFERENCE 2024

The Proceeding of 29th Multidisciplinary Academic Conference, Hummingbird Publications and Research International on African Creativity towards Sustainable Development Goals: Multidisciplinary Approach. Vol. 29 No. 1, 29th-31st January, 2024 at Federal University, Lokoja, Kogi State, Nigeria.

$$\text{Correl}(X, Y) = \frac{\sum(x - \bar{x})(y - \bar{y})}{\sqrt{\sum(x - \bar{x})^2 \sum(y - \bar{y})^2}} \quad (1)$$

Regression Models

The Regression analysis analyzes how a single dependent variable is affected by the values of one or more independent variables. The results can be used to predict the performance of a new untested data set. Bivariate analysis, where exactly two measurements are made on each observation was used. Linear, logarithmic, inverse, quadratic, cubic, compound, power, sinusoidal, growth, exponential and logistic regression models were used to generate metrics with time spent on error as dependent variable $EY(t)$ and number of bugs as independent variable t for each of the pair and individual programmers. This is to check the best fit model. The model metric Equations 2 to 12 are shown (Ajiboye *et al.*, 2023).

Linear Model

$$EY(t) = \beta_0 + \beta_1 t \quad (2)$$

Logarithmic Model

$$EY(t) = \beta_0 + \beta_1 \ln t \quad (3)$$

Inverse Model

$$EY(t) = \beta_0 + \frac{\beta_1}{t} \quad (4)$$

Quadratic Model

$$EY(t) = \beta_0 + \beta_1 t + \beta_2 t^2 \quad (5)$$

Cubic Model

$$EY(t) = \beta_0 + \beta_1 t + \beta_2 t^2 + \beta_3 t^3 \quad (6)$$

Compound Model

$$EY(t) = \beta_0 \beta_1 t \quad (7)$$

Power Model

$$EY(t) = \beta_0 t \beta_1 \quad (8)$$

Sigmoidal Model

$$EY(t) = \exp\left(\beta_0 + \frac{\beta_1}{t}\right) \quad (9)$$

Growth Model

$$EY(t) = \exp(\beta_0 + \beta_1 t) \quad (10)$$

FULOKOJA-2024

FEDERAL UNIVERSITY, LOKOJA, KOGI STATE, NIGERIA-CONFERENCE 2024

The Proceeding of 29th Multidisciplinary Academic Conference, Hummingbird Publications and Research International on African Creativity towards Sustainable Development Goals: Multidisciplinary Approach. Vol. 29 No. 1, 29th-31st January, 2024 at Federal University, Lokoja, Kogi State, Nigeria.

language used in the design of the agile software are shown in Tables 3. The selected model (one with the highest R squared values) is shown in Figures 3.

Table 3: Curve fit models of the time spent by individual expert in debugging error codes in agile software written in JavaScript

Equation	Model Summary					Parameter Estimates			
	R Square	F	df1	df2	Sig.	Constant	b1	b2	b3
Linear	0.910	40.612	1	4	0.003	12.389	13.914		
Logarithmic	0.961	97.334	1	4	0.001	-6.615	57.443		
Inverse	0.935	57.783	1	4	0.002	127.525	-197.841		
Quadratic	0.980	75.405	2	3	0.003	-33.451	37.716	-2.645	
Cubic	0.996	158.558	3	2	0.006	30.705	-14.528	10.018	-0.938
Compound	0.868	26.383	1	4	0.007	26.692	1.239		
Power	0.954	83.281	1	4	0.001	19.413	0.903		
Sigmoid	0.969	125.762	1	4	0.000	5.092	-3.175		
Growth	0.868	26.383	1	4	0.007	3.284	0.214		
Exponential	0.868	26.383	1	4	0.007	26.692	0.214		

Cubic model had the highest R squared values and was selected as the best fit model for agile software written in JavaScript programming language and is represented in Figure 5.

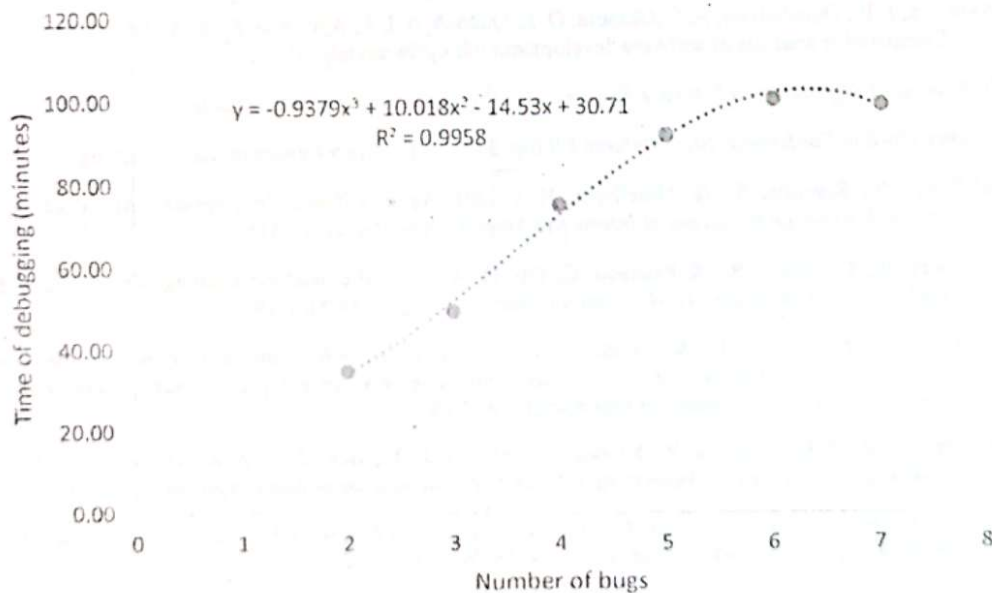


Figure 5: Cubic model of the time spent by individual expert in debugging error codes in agile software written in JavaScript

FULOKOJA-2024

FEDERAL UNIVERSITY, LOKOJA, KOGI STATE, NIGERIA-CONFERENCE 2024
The Proceeding of 29th Multidisciplinary Academic Conference, Hummingbird Publications and
Research International on African Creativity towards Sustainable Development Goals: Multidisciplinary
Approach. Vol. 29 No. 1, 29th-31st January, 2024 at Federal University, Lokoja, Kogi State, Nigeria.

- M.A. Ajiboye, M.S. Abolarin and J.A. Ajiboye (June, 2016), *A Comparison of Code Maintainability in Agile Environment*, The IUP Journal of Information Technology, Vol. XII, No. 2, Pages 33-43.
- Manurung, A. H., & Kurniawan, R. (2022). *Organizational agility: do agile project management and networking capability require market orientation?*, International Journal of Managing Projects in Business, 15(1), 1-35.
- Mantovani Fontana, R., & Marczak, S. (2020). Characteristics and challenges of agile software development adoption in Brazilian government. *Journal of technology management & innovation*, 15(2), 3-10.
- Marion, T. J., & Fixson, S. K. (2021). *The transformation of the innovation process: How digital tools are changing work, collaboration, and organizations in new product development*. *Journal of Product Innovation Management*, 38(1), 192-215.
- Núñez, J. C. S., Lindo, A. C., & Rodríguez, P. G. (2020). A preventive secure software development model for a software factory: a case study. *IEEE Access*, 8, 77653-77665.
- Pettersen, E. F., Goddard, T. D., Huang, C. C., Meng, E. C., Couch, G. S., Croll, T. I., ... & Ferrin, T. E. (2021). UCSF ChimeraX: *Structure visualization for researchers, educators, and developers*. *Protein Science*, 30(1), 70-82.
- Rahanu, H., Loveday, J., Siakas, E., Georgiadou, E., Siakas, K., & Ross, M. (2022). The proposal of adding a society value to the software process improvement manifesto. In *European Conference on Software Process Improvement*, 673-687. Cham: Springer International Publishing.
- Sen, S., Patel, M., & Sharma, A. K. (2021). Software Development Life Cycle Performance Analysis. In *Emerging Trends in Data Driven Computing and Communications: Proceedings of DDCToT 2021* (pp. 311-319). Springer Singapore.
- Zhang, J., Long, J., & von Schaeuwen, A. M. E. (2021). *How does digital transformation improve organizational resilience?—findings from PLS-SEM and fsQCA*. *Sustainability*, 13(20), 11487.

FULOKOJA-2024

FEDERAL UNIVERSITY, LOKOJA, KOGI STATE, NIGERIA-CONFERENCE 2024

The Proceeding of 29th Multidisciplinary Academic Conference, Hummingbird Publications and Research International on African Creativity towards Sustainable Development Goals: Multidisciplinary Approach. Vol. 29 No. 1, 29th-31st January, 2024 at Federal University, Lokoja, Kogi State, Nigeria.

CONCLUSION

In this comprehensive analysis of regression models, we explored various mathematical relationships to understand their efficacy in explaining the variability within the dataset. Each model, ranging from linear to more complex forms such as cubic and logarithmic, was evaluated based on statistical metrics and practical interpretability.

The cubic model emerged as particularly noteworthy, boasting an impressive R Square value of 0.996. This indicates an exceptional fit to the data, showing its ability to explain nearly all the variability observed in the dependent variable. The model's statistical significance, as indicated by a low p-value of 0.006, further underscores its reliability in capturing complex relationships within the dataset.

Additionally, the selection of the optimal model should consider the overarching goals of the analysis. Whether the emphasis lies on predictive accuracy, theoretical alignment, or uncovering intricate patterns within the data, the chosen model should align with the specific objectives of the study.

In conclusion, the cubic model demonstrates robust statistical performance, making it a compelling choice for capturing complex relationships within the dataset. However, the ultimate selection should be a judicious balance between statistical metrics, practical interpretability, and alignment with the study's objectives. As with any modeling exercise, ongoing validation and scrutiny are recommended to ensure the chosen model effectively serves the study's goals and accurately represents the underlying patterns in the data.

References

- Akinsola, J. E., Ogunbanwo, A. S., Okesola, O. J., Odun-Ayo, I. J., Ayegbusi, F. D., & Adebisi, A. A. (2020). Comparative analysis of software development life cycle models (SDLC). In *Intelligent Algorithms in Software Engineering: Proceedings of the 9th Computer Science On-line Conference 2020, Volume 1 9* (pp. 310-322). Springer International Publishing.
- Al-Saqqa, S., Sawalha, S., & AbdelNabi, H. (2020). Agile software development: Methodologies and trends. *International Journal of Interactive Mobile Technologies*, 14(11).
- Ashmore, R., Calinescu, R., & Paterson, C. (2021). Assuring the machine learning lifecycle: Desiderata, methods, and challenges. *ACM Computing Surveys (CSUR)*, 54(5), 1-39.
- Kroener, I., Barnard-Wills, D., & Muraszkiwicz, J. (2021). *Agile ethics: an iterative and flexible approach to assessing ethical, legal and social issues in the agile development of crisis management information systems. Ethics and Information Technology*, 23, 7-18.
- Kuhmann, M., Tell, P., Hebig, R., Klünder, J., Münch, J., Linssen, O., ... & Richardson, I. (2021). What makes agile software development agile?. *IEEE transactions on software engineering*, 48(9), 3523-3539.
- Lawal, A., & Ogbu, R. C. (2021). A comparative analysis of agile and waterfall software development methodologies. *Bakolori Journal of General Studies*, 11(2), 1-2.
- Mary Adebola Ajiboye, Matthew Sunday Abolarin, Johnson Adegbeniga Ajiboye, Abraham Usman Usman, Sanjay Misra, "Pair Programming – Cubic Prediction Model Results for Random Pairs and Individual Junior Programmers." *Engineering World*, vol. 5, pp. 163-166, 2023, DOI:10.37394/232025.2023.5.18

FULOKOJA-2024

FEDERAL UNIVERSITY, LOKOJA, KOGI STATE, NIGERIA-CONFERENCE 2024
 The Proceeding of 29th Multidisciplinary Academic Conference, Hummingbird Publications and
 Research International on African Creativity towards Sustainable Development Goals: Multidisciplinary
 Approach, Vol. 29 No. 1, 29th-31st January, 2024 at Federal University, Lokoja, Kogi State, Nigeria.

Power Model

$$+ EY(t) = \beta_0 e^{\beta_1 t} \quad (11)$$

Logistic Model

$$EY(t) = (1 + \beta_0 \beta_1 t)^{-1} \quad (12)$$

Where

$EY(t)$ is the dependent variable, which is the time spent in debugging (min)

t is the independent variable, which is the number of bugs

$\beta_0, \beta_1, \beta_2, \beta_3$ are coefficients of variables

RESULTS

The average time spent by Individual Expert programmers in debugging a number of error codes (bugs) in agile software written in JavaScript programming language is presented in Table 4.1. The data suggests that as the number of bugs increases, the time required for debugging also increases, demonstrating a positive correlation between these variables.

The relatively consistent standard deviations across different bug counts suggest a degree of stability in the time spent by Individual Expert programmers, indicating a certain level of reliability in their debugging efforts.

The observed pattern provides insights for project managers, indicating that as the complexity of the software increases, additional time may be required for debugging. This information can be crucial for resource planning and setting realistic timelines in agile software development projects led by Individual Expert programmers.

Table 4.1: Average time spent in debugging error codes in agile program by Individual Expert JavaScript programmers

Paired groups	2 Bugs	3 Bugs	4 Bugs	5 Bugs	6 Bugs	7 Bugs
Individual Expert	35.00 ± 7.071 ^a	49.41 ± 11.440 ^b	75.19 ± 26.366 ^b	91.67 ± 32.258 ^b	100.00 ± 31.623 ^b	98.75 ± 11.260 ^b

Means are ± Standard error

Values on the same column with different superscripts are significantly different ($p \leq 0.05$)

Predictive model for the time spent in debugging error codes in software for the use of software project managers

The curve fit regression modelling in SPSS contain linear, logarithmic, inverse, quadratic, cubic, compound, power, sigmoid (S), growth and exponential models that enables the selection of the model of best fit using parameters like the R square values. The curve fit regression models for Individual Expert programmers for JavaScript programming

FULOKOJA-2024

FEDERAL UNIVERSITY, LOKOJA, KOGI STATE, NIGERIA-CONFERENCE 2024
The Proceeding of 29th Multidisciplinary Academic Conference, Hummingbird Publications and Research International on African Creativity towards Sustainable Development Goals: Multidisciplinary Approach. Vol. 29 No. 1, 29th-31st January, 2024 at Federal University, Lokoja, Kogi State, Nigeria.

STATISTICAL ANALYSIS

The data collected from the experiment were subjected to data analysis using IBM SPSS 23.0, 2015 version. One-way variance analysis (oneway ANOVA) was used to check if there are significant differences in the time spent on debugging a number of error codes (bugs) introduced into the developed agile software between the different pair groups of programmers. Where there were significant differences, Duncan Multiple Range Test (DMRT) was used to separate significant means into homogenous subsets.

STATISTICAL TOOLS

The Analysis of Variance (ANOVA) provide different types of variance analysis and performs a simple analysis of variance on parametric data which is drawn from a known population for three or more samples. In this work, ANOVA was used to compare the average time spent on an error and the time spent on each of the project containing different number of bugs between the different pair programmers and the different individual programmer. Duncan Multiple Range Test (DMRT) was used to separate significant means. DMRT is sensitive and used for separation of means within the range of 3 and 7 samples.

Duncan's multiple range test is a commonly used method for comparing all pairs of means (1955). When using Duncan's Multiple Range Test (DMRT), numerical bounds are computed to enable the difference between any two treatment means to be classified as significant or non-significant. A sequence of values that each relate to a particular set of pair comparisons must be computed for DMRT. The standard error of the mean difference is the main determinant. The estimate of variance of an estimated elementary treatment contrast through the design can be used to quickly calculate this. Applying the DMRT, rate each treatment method according to the character being studied, either decreasing or rising.

Correlation Coefficient

The degree to which two measures, let's say X and Y, vary together is shown by the correlation coefficient. Each measurement pair is examined using correlation analysis to see if there is a tendency for the two to move together. The value of the correlation coefficient might range from -1 to +1. When there is a tendency for large values of one variable to be correlated with large values of another, this is known as positive correlation. When one variable tends to have small values and the other to have high values, there is a negative correlation; when both variables tend to have unrelated values, there is a correlation close to 0 (zero). Bivariate correlation given in Equation 1, was used to check the relationships between the number of bugs in projects and the time spent to correct the errors. Bivariate shows relationship between two variables.

FULOKOJA-2024

FEDERAL UNIVERSITY, LOKOJA, KOGI STATE, NIGERIA-CONFERENCE 2024
The Proceeding of 29th Multidisciplinary Academic Conference, Hummingbird Publications and
Research International on African Creativity towards Sustainable Development Goals: Multidisciplinary
Approach. Vol. 29 No. 1, 29th-31st January, 2024 at Federal University, Lokoja, Kogi State, Nigeria.

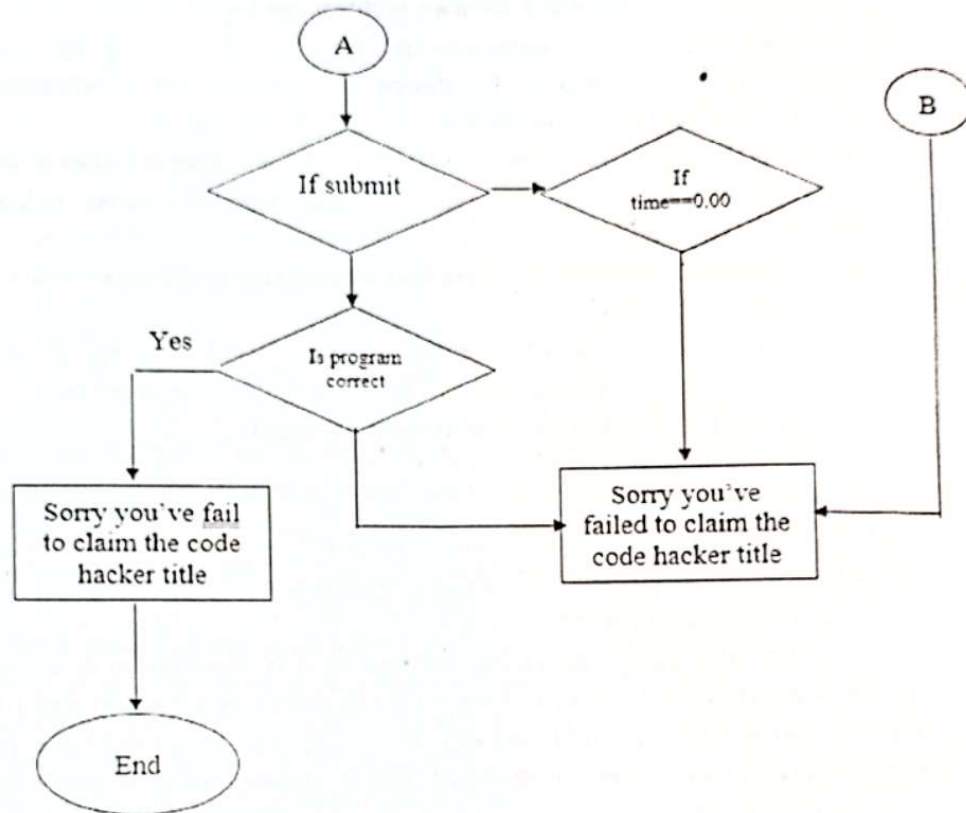


Figure 3: Flow Chart of the Developed Program in JavaScript

LOGICAL DESIGN

By putting the main elements of the system into practice, the logical design transforms the specification of the system requirements into a system model. This layout offers a method via which a user will be monitored during the period of code debugging which allows the Project Manager to measure the productivity metrics of a software developer.

GROUPINGS FOR SOFTWARE PROGRAMMERS

According to Ajiboye *et al.* (2016), the basis of pair composition is programmer expertise which is based on previous experience of pair programming. An Expert programmer is defined as one who has at least five years of experience within the project domain while a Junior programmer is defined as one who has less than five years of experience within the project domain. In this research work, programmers were grouped to work as pairs, where two programmers work together on a task and individuals. The individual experts were chosen with pairing based on their knowledge and experience of pair programming as shown in Table 1.

FULOKOJA-2024

FEDERAL UNIVERSITY, LOKOJA, KOGI STATE, NIGERIA-CONFERENCE 2024

The Proceeding of 29th Multidisciplinary Academic Conference, Hummingbird Publications and Research International on African Creativity towards Sustainable Development Goals: Multidisciplinary Approach, Vol. 29 No. 1, 29th-31st January, 2024 at Federal University, Lokoja, Kogi State, Nigeria.

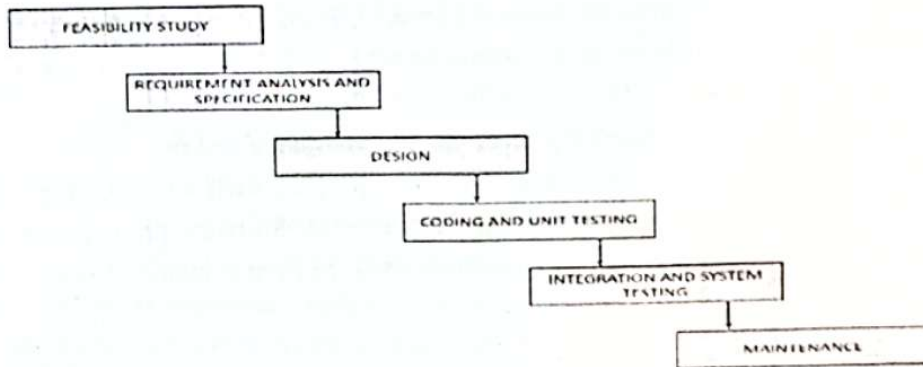


Figure 1: The Waterfall Software Development Process Source: Techtarget (2021)

As a result, a strategy for implementing agile approaches is suggested to account for the flexibility that the waterfall model does not provide. Recently, there have been developments in the software development sector. Agile software development has become well-known and widely accepted as a result. Due to the present software industry's unpredictability, flexibility in software development is absolutely necessary. Due to the dynamic nature of the client's requirements, the conventional approach to software development is insufficient. According to Sen *et al.* (2021); Agile approaches are a combination of iterative, incremental, self-organizing, and emergent software development procedures, as indicated in Figure 2.

Additionally, it promotes and places an emphasis on strong collaboration and relationships between software clients and software developers. Every firm strives to use a software development methodology that produces faster, better, and less expensive solutions (Al-Saqqa *et al.*, 2020; AbdelNabi, 2020). The engagement and regular communication between the development team and stakeholders, as well as the quick delivery of usable products, are at the heart of the agile methodology's problems.

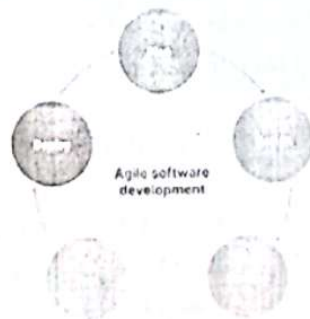


Figure 2: Agile Methodology Source: Ahal Labs

From Figure 2 the agile methodology is;

FULOKOJA-2024

FEDERAL UNIVERSITY, LOKOJA, KOGI STATE, NIGERIA-CONFERENCE 2024

The Proceeding of 29th Multidisciplinary Academic Conference, Hummingbird Publications and Research International on African Creativity towards Sustainable Development Goals: Multidisciplinary Approach. Vol. 29 No. 1, 29th-31st January, 2024 at Federal University, Lokoja, Kogi State, Nigeria.

(2022), Prashar (2021), and Shastri *et al.* (2021), the three key restrictions are money, time, and scope.

Software engineering circles have discussed on the best way to organize software development to deliver quicker, more effective, and less expensive solutions. Many There have been recommendations made for improvements. Along with the standardization and assessment of the software process, this comprises a wide range of valuable tools, methodologies, and practices (Pettersen and Godderd, 2021). Most enhancement proposals have come from software specialists who have separately built their own methods and procedures to deal with the anticipated shift.

A group of seventeen software developers got together in Utah in February 2001 to talk about the problems the software industry was facing and came up with the agile manifesto. The phrase "agile methods" describes a collection of various techniques and procedures that have similar goals and core principles. The traditional plan-based strategy, which focuses primarily on an efficient, well-organized, and logical engineering-based approach, is really what these agile development strategies are in response to (Rahanu *et al.*, 2022; Recchioni and Racinelli, 2022).

The traditional method of software development allows for a complete specification of difficulties. With careful planning, documented procedures, and extremely thorough and diligent reuse of codes, every challenge is given optimal and predictable answers, which makes the development activity effective. The agile software development process, on the other hand, approaches the problem of an unpredictable world by putting more of an emphasis on people and their ingenuity than on processes (Kuhrmann *et al.*, 2021; Linssen and Richardson, 2021).

An iterative method of finishing a project during the course of its existence is called agile project management. Multiple iterations or incremental steps toward the completion of a project make up iterative or agile life cycles. Iterative techniques, which allow you to make changes as you go instead of following a linear path, are frequently employed in software development projects to increase stability and adaptability (Zhang *et al.*, 2021; Von, 2021). The goal of an agile or iterative strategy is to make sure that benefits are released frequently throughout the process rather than just at the conclusion. Autonomy, respect, and adaptability are three attitudes and behaviors that agile projects should fundamentally encourage. The project managers are who? In a range of businesses, project managers use their project management expertise to initiate, carry out, and complete projects (Kroener *et al.*, 2021; Barnard-Wills and Muraszkievicz, 2021). Software project managers are responsible for managing the planning, scheduling, budgeting, execution, and delivery of software and web projects. They oversee the team members working on the projects and ensure that every software project is effectively finished.

They are the creators of some of the most cutting-edge contemporary products, services, and methodologies, from smartphone applications to the lavish buildings of foreign towns. Project managers possess a set of abilities that help them do each task more quickly and