# SYSTEM PERFORMANCE METRICS IN AN AGILE COMPUTING ENVIRONMENT (A CASE STUDY OF INTEGRATED DATA SERVICE LIMITED)

**ODIKWA, H.N.[1]*, UZOARU, G.C.[2] AND OKPAKO, E.[3]**
[1] Department of Computer Science, Abia State University, Uturu, Nigeria
[2] Department of Information Technology, Clifford University, Owerrinta, Nigeria
[3] Department of Computer Science, Edwin Clark University, Nigeria

## ABSTRACT
This paper is aimed at determining the best system performance metrics that will quickly respond to changes in an agile computing environment. This paper designed an architectural model for determining the best performance metrics in an agile computing environment such as Integrated Data Services Limited. The research work was based on six (6) system performance metrics; execution time, QUIPS, SPECs, Clock rate, MFLOPS and MIPS. MATLAB as a mathematical tool was used in the analysis of the various performance metrics to determine the best metrics. The experiments performed showed that execution time is the best performance metrics having execution time of 0.95, 0.97 and 1.01 followed by QUIPS and SPEC that has values of 6, 3 and 1.2 seconds in completing processes in the three experimental servers, while MFLOPS, MIPS and Clock Rates were found to be of poor performance metrics in any agile computing environment.

**KEYWORDS**: Agile computing, Extreme programming, Performance metrics, Scrum.
***Correspondence:** odikwa@yahoo.com.

## INTRODUCTION

System performance is the effectiveness with which the resources of the host computer system are utilized toward meeting the objectives of the software system [1]. Performance metrics are the foundation of experimental computer science and engineering. Obviously, adequate metrics are of primary importance to research progress, whereas inappropriate metrics can drive research and development in wrong or unfruitful directions. The recent trend toward multi-core and many-core systems makes adequate performance metrics for assessing the performance of multithreaded computer systems running multiple independent programs essential. Researchers have reached the consensus that the performance metric of choice for assessing a single program's performance is its execution time [3]. The Agile development movement started in earnest in the 1990s as a rejection of the establishment with its rather staid and seemingly sluggish development methods known generally by names such as the waterfall model or V-model [1, 2]. The Agile Manifesto is a document that is discussed and argued about a lot. The argumentation is partially related to how the Agile Manifesto is understood (or not understood) and whether people agree or do not agree about it. Partially because of this, a non-profit organization called the Agile Alliance was formed in 2001 to promote the principles and values listed in the Agile Manifesto [1].

Integrated Data Services limited is an arm of the Nigerian National Petroleum Corporation that is saddled with the responsibility of handling some of the vital data storage and transmission of the parent organization. It is an agile computing system having a network distributed system. The organization is faced with enormous task of finding the best performance metrics in terms of system evaluation that suits the tasks of their operations. This task will help improve the performance of the organizational capacity in terms of efficiency and reliability of their server systems to give optimal performance. Clients complaints have necessitated to investigating a performance metrics best suitable for carrying out operations for both the existing systems and yet to acquire systems. Performance is the degree to which any system meets expectations of the person involved in using it [6, 10], used the term 'performance' to signify how sound a system assumed to perform correctly during process execution. Factors that can affect system performance metrics are; bandwidth and latency as shown in figure 1 and figure 2.

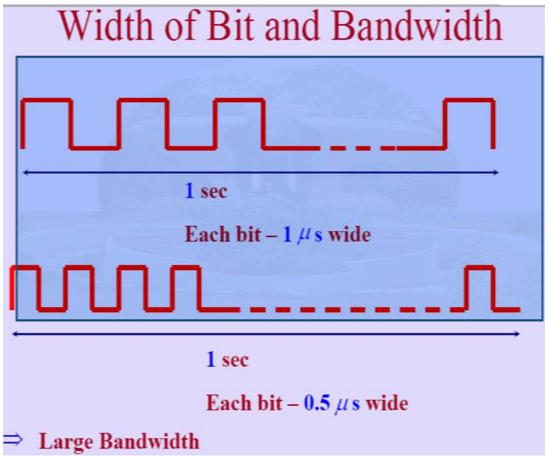


**Figure 1**: Width of bit and bandwidth [14]

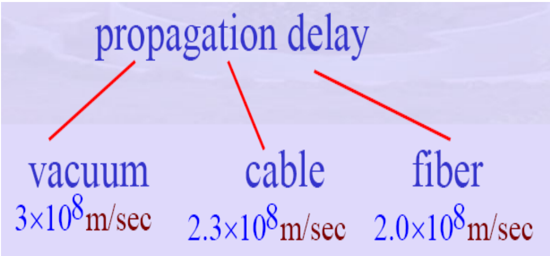


**Figure 2**: Propagation delay in network media [14]

Integrated data service as a company uses scrum which is defines as a flexible, holistic product development strategy where developers work as a unit to reach a common goal [7]. In Scrum, iteration is called sprint, with a usual duration from one week to one month, however, at the beginning of project, sprint planning is initiated to specify and prioritize the features as depicted in figure 3.

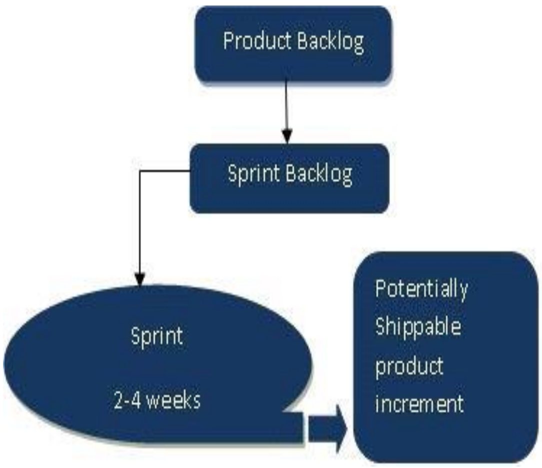


**Figure 3:** Scrum methodology [7]

In addition, the integrated data services also employed the agile method of eXtreme programming (XP) which has features such as continuous concrete feedback from short cycles of development, face-to-face communication, simplicity in design, adaptive approach to accommodate the change in business requirement, automated testing process and must identify and correct the wrong going processes [5, 8]. Software development is based on an incremental, iterative approach in data integrated services as shown in figure 4. Agile methodologies have the possibility to adapt to changing requirements over time and to have continuous feedback from the end users [15].

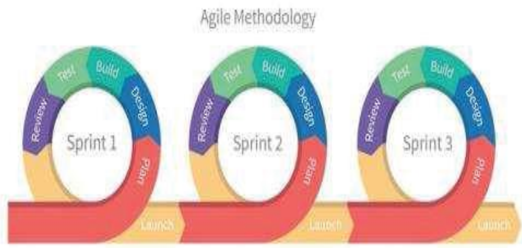


**Figure 4:** Agile methodology [8]

The traditional software development process, for example, waterfall model follows defined phases which occur in sequence [8].
According to human resource researchers, 70% of Multi-National Companies are moving away from the outdated annual review approach to performance management [9]. Delayed correction is another factor leading to the in-efficiency of annual performance appraisal. Due to annual review, employees might be heading in the wrong direction. With regular review of employee performance, timely corrective action would be possible [9].

## MATERIALS AND METHODS

### Study area and sites

Integrated Data Services Limited is located at Benin City in Edo State of Nigeria which is in the south-south geopolitical region of Nigeria.

### Study design

The figure 5 shows an extension of the system performance metrics model architecture adopted from Integrated Data Services Limited. In order to ascertain and measure system performance metrics in an agile computing environment and to determine the best and optimal performance, three server machines from Integrated Data Service (IDS) Benin City were engaged in this study. Each of the servers has 20 (twenty) clients attached to it. The performance metrics were carried out on MATLAB (Mathematics Laboratory) using the execution time, clock rate, MIPS, QUIPS, SPEC and MFLOPS where projects are run in a sequential cycle. It follows a fixed sequence: initiation, planning, execution, monitoring, and closure.

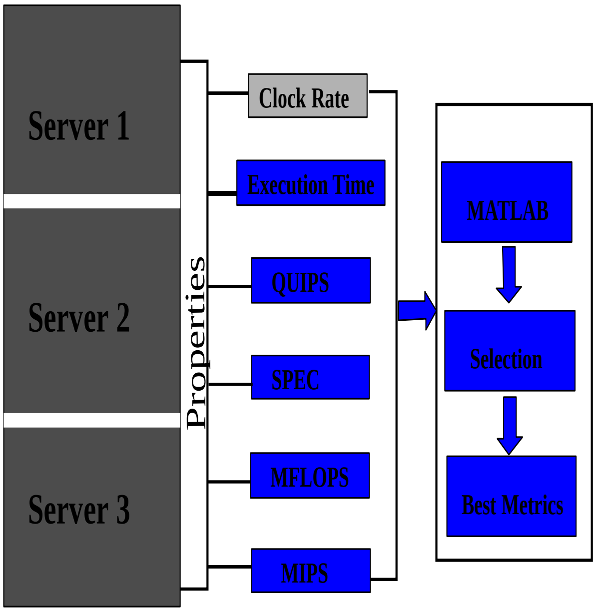


**Figure 5**: System performance metrics architecture

**Sample collection and analysis**

Three servers, A, B and C from Integrated Data Service, Benin in Nigeria were deployed to provide sixty (60) clients with resource information to carry out day-to-day operations with different capacities and capabilities to ascertain the best performance metrics in the computing environment. Table 1 depicts the contents of the three (3) servers in the organization. Variations are observed in terms of processors, RAM size, hard disks and central processing unit.

**Data analysis**

This research work employed many system performance metrics in determining the best metrics for measuring performances in an agile computing environment based on the properties listed in table 1.

**Execution Time**

This is the time taken to execute a program from the beginning to the end. The three servers were made to perform same task at the same time with equal number of clients logged in to use system resources from the server. The execution time was determined by employing equation 1.

$$CPUexecutionTime = CPUClockCycles \cdot ClockCycle \quad (1)$$

$$CPUexecutionTime = NI \cdot CPI \cdot CYT \quad (2)$$

The equation (2) expanded the CPU clock cycles into NI as the number of instruction and CPI as the clocks per instruction and then combined it with CYT the clock cycle time

$$CCT = 1 / CCR \quad (3)$$

The equation (3) describes the clock cycle time as the inverse of CCR which is the clock cycle rate. Equation (3) resulted into equation (4) by considering the CPU clock cycles of any computer system.

$$CCC = No.Inst \cdot CPI \quad (4)$$

CCC is the CPU clock cycles, CPI is the clock per instruction. This gives the general formula given in equation (5) used for calculating the computer system's execution time.

$$CPUClockCycles = \sum_{i=1}^{n}(\text{CPI} * \text{Ci}) \quad (5)$$

Where CPI $_I$ = CPI for a particular class of instructions

C $_i$ is the number of instruction of the ith class that have been executed.

**Clock rate**

The clock rate described the frequency during which a clock circuit of a system's processor generates pulses which are deployed to synchronize the general operations of its components and is also used as a good indicator of processor's speed. Equation 3 determines the clock cycle time, while equation

$$CPUTime = No.CC * CCT \quad (6)$$

Equation 6 is further expanded to equation 7.

$$CPUTime = No.CC / CR \quad (7)$$

CC is the clock cycles, CR is the clock rate

The clock rate is determined generally with equation 8

$$ClockRate = No.CC / CPT \quad (8)$$

Where CPT is the CPU time in seconds.

**MIPS (Millions of Instructions Per Second)**

$$OverallMIPS = \frac{\left(\sum_{i=0}^{n} I_i\right)}{\left(\sum_{i=0}^{n} T_i\right)} \quad (9)$$

$I_i$ is the instruction count of $_i$th bench, $T_i$ is the execution time of ith benchmark.

To calculate the weighted arithmetic mean equation 10 is employed.

$$WAM = \frac{1}{\left(\sum_{k=1}^{n} t_k\right)}\left[\sum_{i=1}^{n} t_i MIPS\right] \quad (10)$$

By substituting equation (9) in (10) gives equation 11

$$WAM = \frac{1}{\left(\sum_{k=1}^{n} t_k\right)}\left[\sum_{i=1}^{n} t_i \frac{I_i}{t_i}\right] \quad (11)$$

Finally, by cancelling out the ti's, the weighted arithmetic mean calculations for the weights of the three server systems is given in equation 12.

$$WAM = \frac{\left(\sum_{i=1}^{n} I_i\right)}{\left(\sum_{k=1}^{n} t_k\right)} \quad (12)$$

Where $I_i$ is the instruction count and $t_k$ the time.

**MFLOPS**

Mflops defines the unit of distance traveled by a computer system when executing a program. It defines an arithmetic operation on two floating point ~~(i.e~~ fractional) quantities to be the basic unit of distance.

$$Mflops = \frac{f}{\left(t_c \times 10^6\right)} \quad (13)$$

Equation 13 is to determine millions of floating-point operations executed per second. F is the number of floating-point operations executed in $t_c$ seconds.

**SPEC (Standard Performance Evaluation Corporation)**

The SPEC identifies a set of integer and floating-point benchmark program that is intended to reflect the way the workstation class in Integrated Data Services Limited computer systems were actually used or being used. They reflect standardized methodology for measuring and reporting the performance obtained when executing programs.

**SPEC Algorithm**

1. Measure the time needed to execute/accomplish each program in the set of three server systems.
2. Then, divide the time measured in step 1 for each of the program on a standard basis to normalize the execution times.
3. Average together all the normalized values using the geometric mean.

To determine SPEC in the three servers, equation 14 is employed which is given as

$$SPEC = \frac{\left(\sum_{t=1}^{n} T_e\right)}{\left(\sum_{t=1}^{n} T_n\right)} \quad (14)$$

$T_e$ is the execution time and $T_n$ is the normalized execution time.

**QUIPS (Quality Improvement per Second)**

The QUIPS define the quality of the solution as a more significant indication of a client's final goal. The quality is meticulously defined on the basis of mathematical characteristics of the problem being solved. Dividing this measure of solution quality by the time required to achieve that level of quality produces Quips.

$$QUIPS = TW \,/\, TE \quad (15)$$

The QUIPS depicted in equation (15) determines the total number of work against the time of execution.

**RESULTS**

Experiments were carried out using different server properties depicted in table1. The system performance metrics were carried out using the following metrics such as clock rate, SPEC, execution time, MIPS, QUIPS and MFLOPS where the analysis were done in MATLAB.

**The clock rates**

**Table 2:** Experiments with clock cycle time and CPU clock cycles

| Attribute | Server 1 | Server 2 | Server 3 |
|---|---|---|---|
| Clock Cycle Time | 10 seconds | 15 seconds | 25 seconds |
| CPU Clock Cycles | 20 seconds | 30 Seconds | 40 seconds |

**Execution time**

**Table 3:** Experiments with number of instructions, average cycles and clock cycles time.

| Attribute | Server 1 | Server 2 | Server 3 |
|---|---|---|---|
| Number of instructions in program (I) | 3 | 3 | 3 |
| Average cycles per instruction (CPI) | 4.5 | 4.6 | 4.8 |
| CPU Clock cycles | 13.50 | 13.80 | 14.40 |
| Clock Cycle Rate | 0.07 | 0.07 | 0.07 |
| Execution Time (secs) | 0.95 | 0.97 | 1.01 |

**MIPS**

Employing equation (9), the various overall MIPs are determined

$$overallMIPs1 = \frac{700}{2} = 350$$

$$OverallMIPs2 = \frac{200}{4} = 50$$

$$OverallMIPs3 = \frac{1000}{5} = 200$$

The weights of the benchmarks with respect to instruction counts:-

$$\left\{\frac{700}{1900}, \frac{200}{1900}, \frac{1000}{1900}\right\} = \left\{0.37, 0.11, 0.53\right\}$$

Weight of benchmarks with respect to time:-

$$\left\{\frac{2}{11}, \frac{4}{11}, \frac{5}{11}\right\} = \left\{0.18, 0.36, 0.45\right\}$$

**Table 4:** calculation of MIPs with respect to three servers

| Benchmark | Instruction Count (in millions) | Time (secs) | Individual MIPs |
|---|---|---|---|
| 1 | 700 | 2 | 350 |
| 2 | 200 | 4 | 50 |
| 3 | 1000 | 5 | 200 |
| Total | 1900 | 11 | 600 |

Table 4 shows the calculations of overall MIPs of the three servers in determining the best approach to system performance metrics.

Weighted harmonic mean (WHM) of individual MIPS (weighted with 1-counts) =

$$\frac{1}{\left(\frac{0.37}{350}+\frac{0.11}{50}+\frac{0.53}{200}\right)}=\frac{1}{(0.00105+0.0022+0.00265)}$$

$$=\frac{1}{0.0059}=169.49$$

Weighted arithmetic mean (WAM) of the individual MIPs (weighted with time) =

$$(350\times 0.18+50\times 0.36+200\times 0.45)=63+18+90=171.$$

**Table 1:** Server Properties of Integrated Data Services

**Name of computerRAMSizeHardDiskProcessorCPU**

HP Proliant ML 10 Gen9: System 124GB1TB3.30GHZE3-1225V5
HP Proliant D380 Gen7: System 216.0GB1TB2.40GHZE5620
HP Proliant ML10 Gen6: System 312.0GB500GB2.40GHZ X3430

**Table 5:** MFLOPS calculations based on floating points

| Server | Floating Point | Te(s) | MFLOPS | Te(s) | MFLOPS |
|---|---|---|---|---|---|
| S1 | $1.45x10^{3}$ | 20 | $7.25x10^{-5}$ | 15 | $9.6x10^{-5}$ |
| S2 | $1.41 \times 10^{3}$ | 20 | $7.05 \times 10^{-5}$ | 15 | $9.4 \times 10^{-5}$ |
| S3 | $1.43 \times 10^{3}$ | 20 | $7.15 \times 10^{-5}$ | 15 | $9.5 \times 10^{-5}$ |

**Table 6:** Results from SPEC with the three servers.

| Program | S1 | S2 | S3 |
|---|---|---|---|
| 1 | 206 | 324 | 367 |
| 2 | 31,240 | 36,484 | 62,204 |
| 3 | 6,300 | 7,240 | 8,240 |
| 4 | 84 | 65 | 56 |
| 5 | 811 | 721 | 421 |
| Geometric mean | 1225 1320 | 1347 | |
| Rank | 1 | 2 | 3 |

Table 6 shows the SPEC calculations normalized using the standard system of S1(server 1) and by employing equation (15), the geometric mean was determined using equation (16).

$$Gm = \sqrt{(x_1 x_{2,} \ldots, x_{1,} \ldots, x_n)} \qquad (16)$$

**Table 7:** Experiments with QUIPS

| Attribute | Server 1Server 2Server 3 |
|---|---|
| Average Number of work303030 | |
| Average Execution Time (secs)51025 | |
| QUIPS (secs)631.2 | |

Table 7 is derived using equation (15)

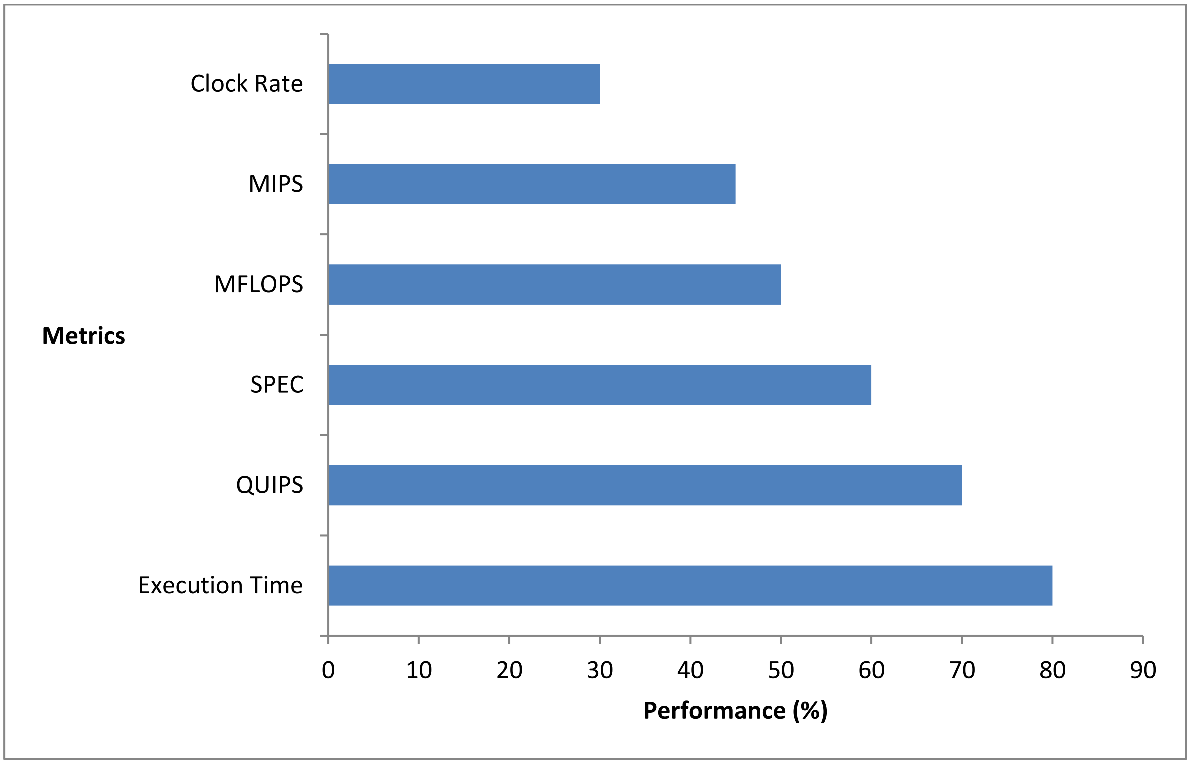


**Figure 5:** Graph showing the system performance metrics

## DISCUSSIONS

The clock rate was determined as shown in table 2 by employing equations 6, 7 and 8. But in reality, the clock rate did not give a clear indication in determining the processor speed in carrying out a particular process schedule in the three systems used in the research work. Table 3 shows the execution time in executing jobs and in this case real time job processing in the organization based on the number of instructions in program and average cycle per instruction which were determined with equations 1, 2, 3 ,4 and 5. System 1 was able to execute a particular job at 0.95 hours followed by system 2 at 0.97 hours and system 3 at 1.01 hours respectively. This could be attributed to the processor size and RAM size of the systems. The MIPS were determined for the three systems as depicted in table 3. The MIPS calculations did not give a clear distinction of which system performed better than the other and therefore is not a very good tool for system performance metrics in an agile computing environment. The weighted mean of the systems were found to be 171. The MFLOPS of the three server systems were determined in table 5 with their various floating points. The result shows that system S2 has the lowest MFLOPS, followed by system 3 and system 1 in an execution time of 20 seconds. In table 6, the result of SPEC shows that with geometric mean of the three systems, system 1 ranked first followed by system 2 and system 3 respectively. When the normalization is done by reordering the process using the standard computer S1, the resultant effect is that S1 performed best in that it produces the shortest execution time. Comparisons of the different metrics were done as depicted in fig 5 which affirms that execution time is the best performance metrics in determining the quality, speed and reliability of computer systems in Integrated Data Services Limited.

## CONCLUSION

The study shows that execution time clearly distinguished the system performances in terms of executing a particular referenced job at a particular point of time and therefore determines the best server system in an agile computing environment. This approach in determining the best performance metrics yielded benefits which includes high quality, more productivity, better business value, fewer costs, and quicker time-to-market speeds in an agile computing environment.

## ACKNOWLEDGMENT

We wish to acknowledge the efforts of Mr. Eze Enemana of Data integrated services Benin City, Edo State.

## REFERENCES

1. BARRY, B. (1981). *Software Engineering Economics*. Aukland, New Zeeland: Prentice Hall
2. COCKBURN, A. (2005). Two Case Studies Motivating Efficiency as “Spendable”

Quantity. *Proceedings of the International Conference on Agility*.
3. EXTREME PROGRAMMING. What is Extreme Programming? Paper presented at Australian Association of Scientific Research Conference, Singapore. Retrieved from www.extremeprogramming.org.
4. HENNESSY, J.L. & PATTERSON, D.A. (2003). *Computer Architecture*: *A quantitative Approach (*3rd ed.). Morgan Kaufmann.
5. JOHN, L.K. (2006). *Aggregating Performance Metrics over a Benchmark Suite, Performance Evaluation and Benchmarking* (2nd ed. pp. 47-58). CRC Press.
6. KEEFE, K. & DICK, M. (2004). Using Extreme Programming in a Capstone Project. *International Journal of Advance Computing*. **5**(30): 151–160.
7. KRISSHNAN, R., LINSEY, R., MICHAEL, M., & AMAZON, K. (2004). Performance Metrics for Manufacturing Programs. *MIT Journal Leaders*. **4**(5): 2-9
8. NIKITINA, N. & KAJKO, M.M. (2011), Developer-driven Big-bang Process Transition from Scrum to Kanban. *Proceedings of the International Conference on Software and Systems Process*, pp. 159–168.
9. ODIKWA, H.N., UGWU, C., & INYIAMA, H. (2017). The Application of Extensive Feature Extraction as a Cost Strategy in Clinical Decision Support. *International Journal of Data Mining and Knowledge Management Process*, **10**(10): 110-125
10. POWELL-MORSELL, A. (2019). System Performance Metrics the way out. Retrieved from www.airbrake.io/blog/sdlc/waterfall-model
11. SHERWOOD, T. & BROWN, J.A. (2009). Multithreading Simulation. *Journal of IEEE*, **10**(5): 26-34.
12. SNAVELY, A. & TULLSEN, D.M. (2000). *Symbiotic Job Scheduling for Simultaneous and Multi- threading Processor*. ACM Press.
13. STALLINGS, W. (2010). *Operating System: Internals and Design Principles*. Prentice Hall.
14. TULLSEN, D.M., & BROWN, J.A. (2001). Handling Long Latency Loads in A Simultaneous Multithreading Processors. *International Journal IEEE*. **2**(10), 318-327.
15. VAN B.M., SHERWOOD, T. & CALDER, B. (2004). A Co-phase Matrix to Guide Simultaneous Multithreading Simulation. *International Journal of IEEE,* **2**(12): 45-56.