International Journal of Mechanical Engineering & Computer Sciences, Vol.1, Issue 1, Jan-Jun, 2017, pp 18-24

www.arseam.com

AN IMPROVED AGILE DEVELOPMENT METHODOLOGY

ALABI O. A[1] NWACHUKWU E.O [2]

[1,2] DEPARTMENT OF COMPUTER SCIENCE

UNIVERSITY OF PORT HARCOURT, NIGERIA

ABSTRACT

One of the major challenges in software development in contemporary time is what methodology to use in the development of application. This is compounded by the multiplicity of software methodologies and even attempt at modifying the classical once. In this project, we analyzed and validated AN IMPROVED AGILE METHODOLOGY using its guiding agile software methodology properties and developed an improved agile methodology. The improved Agile methodology is a useful framework that is being used in solving the problem of documentation we might have been encountering in developing software by the used of JAVA DOC. as a proper storage of our documentation in HTML format and is been compared with existing agile methodology.

Keywords: Documentation, Javadoc, Agile Methodology.

Introduction

When a software methodology is developed a major progress is made. AN IMPROVED AGILE DEVELOPMENT METHODOLOGY is a new agile software engineering methodology that has been developed to be more tolerant with the developers and also capable of delivering working software that is not only efficient and cost effective but also timely to users. This development strategy mandates developers and software engineers to reason along usability, simplicity, resource distribution, time and cost factor realignment, security, patient inert needs and concurrency issues in a balanced path from the requirements and specification of software and its retirement.

However, the question remains, to what extents can this new methodology go in meeting these challenges. In the development of many other software methodologies, processes are simply given as guidelines believing that the developers and users of the methodology will follow the process and arrive at the expected result. In this newly improved agile methodology, it is believed that nothing is left out rather; during planning there must be a measurable target such that the target is reached we can measure degree of success.

This new improved methodology emphasizes state instead of stages in software development cycle and because state is not only transient but also transitional, any flow is not only possible but adoptable to varying times. State is more people and software workability oriented. It focuses on more on the activity to be accomplished than the steps taken to achieve that the new improved agile developed methodology is a novel software process that is introduces for used in modern software development.

Agile software engineering unites intimately a philosophy and a set of development guidelines in establishing working software. Their major emphasis is based on response to change and how customers can collaborate to work together as well as team interactions (Pressman, 2005).

Materials and methods

This project work is based on proper documentation of the user requirements away from the coding, due to agile believe documenting the documentation inside the main source code which makes it difficult for nor-programmers to access the program outside the team to be able to updates and validate any changes from the user even after software delivery to the customer

Moniruzzaman and Hossian (2013); Hneif and Hockow (2009), carried out a comparative study on several agile software development methodologies identifying key benefits of the agile software development methodology (ASDM) over their traditional counterpart (TSDM).

The improved Agile methodology is a useful framework that is being used in solving the problem of documentation we might have been encountering in developing software by the used of JAVA DOC. as a proper storage of our documentation in HTML format and is been compared with existing agile methodology.

Result

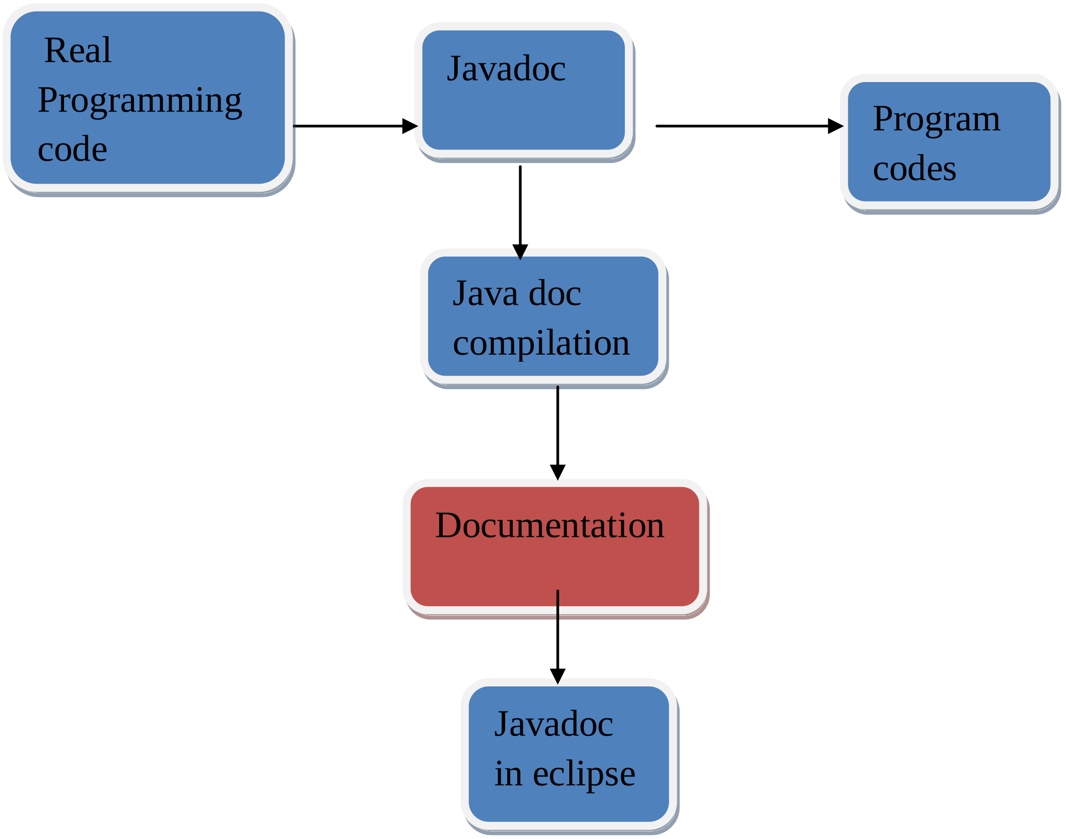


Figure 1: showing how documentation is done

Performance evaluation

Metric	Formula	Significance
Code defect density (in system test)	NDT/KLOC	Indicates the quality of the code presented to the system test group
Code defect density (in operation)	NDO/KLOC	Indicates the quality of the code delivered to the customer
Defect removal efficiency	NDT/(NDT + NDO)	Indicates the effectiveness of the test function in removing defects from the delivered software product

Where:

KLOC = Thousand Lines of Code

NDT = Number of Defects Detected in Testing

NDO = Number of Defects Detected in Operation

Statistical data 1

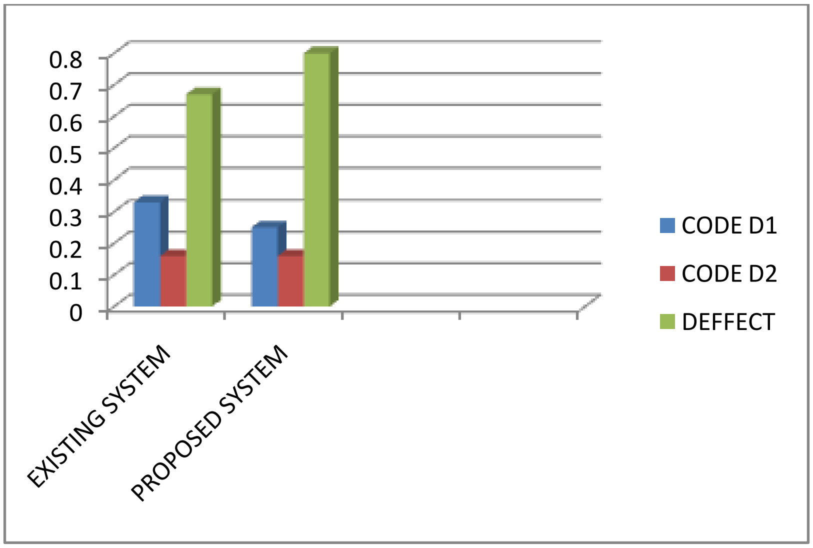
Metric	NDT	NDO	KLOC	EXISTING SYSTEM
Code defect density (in system test)	40		120	0.33
Code defect density (in operation)		20	120	0.16
Defect removal efficiency	40	20		0.67
%				67

Statistical data 2

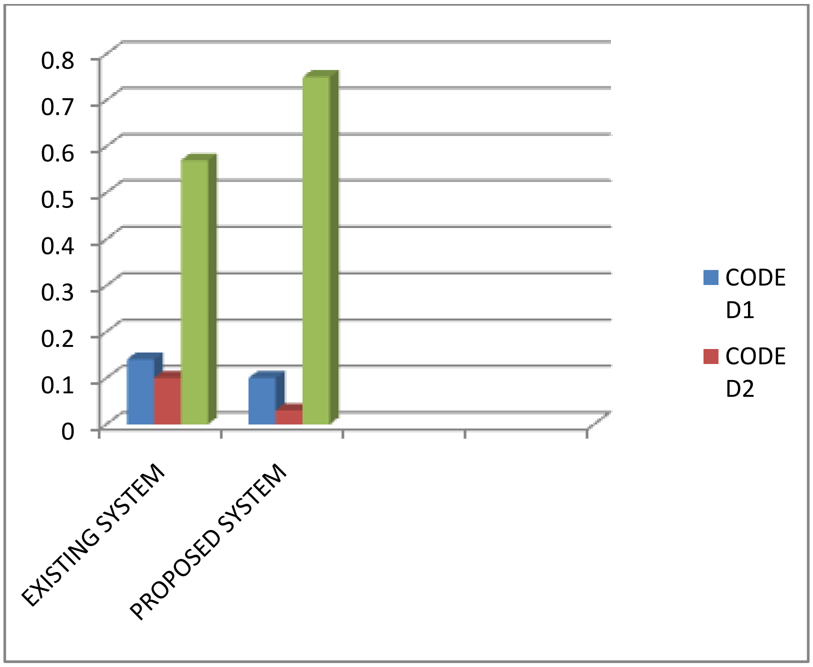
Metric	NDT	NDO	KLOC	PROPOSED SYSTEM
Code defect density (in system test)	30		120	0.25
Code defect density (in operation)		20	120	0.16
Defect removal efficiency	30	20		0.8
%				80

Results

The Graph to Compare both Existing And Proposed System



THE GRAPH OF DATA SET 1



THE GRAPH OF DATA SET 2

The result for selected tasks comparing the agile methodology and our improved agile development software methodology is based on the proper documentation of users requirements in separate modules for easy access and update incase of feedback from the market and speed up the time factor in adjusting the feedback comment.

Discussion

Using the proposed system, the software development effort is improved and the development times halved. This can attributes to the use of information expanders and greatly improved net software development effort. Thus, it is possible for end users to update information remotely and delivery software at adequate time.

REFERENCE

Ambler, S. W. and Jeffries, R. (2002). *Agile Modeling: Effective Practices for Extreme Programming and the Unified Process.* New York: John Wiley & Sons., **3** (2), 25–36

Anita D. Carleton, Robert E. Park and Wolfart B. Goethert, G (1994): The SEI Core Measures The Journal of the Quality Assurance Institute Computer Research Inc. P.O. Box 82266, Phoenix, AZ, USA.

Arisholm, E., Gallis, H., Dyba, T. and Sjoberg, D. I. K. (2007). 'Evaluating Pair Programming with Respect to System Complexity and Programmer Expertise'. *IEEE Trans. on Software Eng.*, **33** (2), 65–86.

Moniruzzaman, A. B. M. And Hossaain, S. A. (2013). Agile Process And Methodologies: A Comparative Study. Global Journal on Computer Sciences and Technology. Vol. 13 Issue 7 (1), 1–25.

Nwachukwu, E.O. and Eke, B.O.(2008). Critical Analysis of Software Development Strategies. Journal of Science and Technology. Vol.7, No. 4. 1-7

Pressman, R. S. (2005). Software Engineering, A Practioners Approach, McGraw- Hill USA, Sixth Edition