The Selection of Programming Language to reduce Defect and increase Quality

Ashwin Tomar, V. M. Thakare

Abstract— There are many different programming languages used for development of various applications. The defects are associated with programming languages. The selection of programming languages depends on user's goal. So variable like defects, STAGES, types of languages and their behavior is studied to find relationship between them. Every language has advantages and disadvantages. It was found that the programming languages are associated with number of defects and hence quality and productivity.

Index Terms— DRE – Defect removal efficiency, DP – defect potential, PL – Programming languages

1 Introduction

THE act of simulating something first requires that a model be developed; this model represents the key behaviors/ functions or characteristics of the selected physical or abstract system or process. The model represents the system itself, whereas the simulation represents the operation of the system There are many programming languages for developing software. Every language has advantages and disadvantages. Selection of programming languages is associated with error or defect number and hence with quality and productivity of software. In earlier classes we studied that languages are classified as Lower level languages like machine languages (uses 0 & 1), and assembly languages (uses instruction as ADD for addition), Middle level language (procedural languages) i.e C and others (Code converts to machine code to run on different machines) and High level languages which includes scripting languages, object oriented languages, (Java, C #, Dot Net) [1].

A software defect is an error, flaw, bug, mistake, failure, or fault in a computer program or system that may generate an inaccurate or unexpected outcome, or precludes the software from behaving as intended [2]. It is nothing but a variance from the given specification, a hidden or coding error. Defects are undesirable, they cause increase in risk, revenue loss to the customer if they remain in the final product. Bad fixes are errors which are not detected and not removed at any stage. So they passed to delivery stage. In this paper an attempt is being made to study the relationship between variable and their behavior. The paper goes in following sequence as introduction, methodology, data collection, result, conclusion and references.

2 Survey

G.Phipps et.al[3] while comparing program of C++ and java experienced that C++ program contained 2-3 times more defects than a Java program, C++ generated 15-30% more defects per line, and Java was 30-200% more productive in lines of code produced over time.

D.K.Verma et.al [4] showed how the design of the language, the form of its specification, and the quality of the Implementation, all have a significant effect on software quality. All these three are the major issues of programming Languages that affects the quality of the software product. Programming languages play an important role in quality of a software product. So it is essential to choose the right programming language for a particular domain to achieve the quality of software product.

P.Bhattacharya et.al. [5] Showed that applications that start with C as the primary language are shifting their code base to C++, and that C++ code is less complex, less prone to errors and requires less effort to maintain.

D.Renu et.al [6] classified the causes of software errors/defects related to requirement, client developer's communication failures deliberate deviation from software requirement, logical design errors, coding errors, non compliance with documents and coding instruction, short coming of testing process, procedure errors, document errors.

3 Methodolgy

In this paper the relationship between variables and their behavior is studied. It is seen that one variable influences or impacts other variable. The variables are classified into independent and dependent variables. Table 1 represents the various STAGES like Requirement, Design, Coding, and Document.

TABLE 1
Open Source And Commercial Tools

Independent Variable	Description
STAGE	Variable represents stage like requirement, design, coding, documents
Programming Language	PL-Variable represents type of language used C, C++, VB

- *Ashwin Tomar is working in computer science, MCA Department, Pune University,Pune, Siddhant Institute of Computer Application, India. Email: mcatomarashwin@gmail.com*
- *V. M. Thakare is working in P. G. Department of CSE, Amravati University, Amravati, India. Email: Vilthakare@gmail.com.*

Data Collection - A good data file from www.namcookanalytics.com reported by Dr Caper Jones was taken as basis. It had 61 cases (0 to 60) of Software Risk Master™ Quality with details on Stage (requirement, design, code, document, bad fixes). Various operations were done on this file to find the behavior of variables and their relationship. So the variable was classified into Independent variable like Language/ Tool and dependent variables like Defects (d). The Table 2.1, 2.2, 2.3 shows relation between various STAGES (like Requirement, Document, Design, Code, and Bad fixes), languages /tools and number of defects.

TABLE 2
AVERAGE DEFECT AT VARIOUS STAGES V_S LANGUAGES

STAGE	ADA	ASM	C	C#	C++
Bad fixes	267.48	614.77	253.570	225.23	125.97
Code	965.31	4881.14	1082.82	670.93	421.02
Design	1267.38	1418.83	1098.41	1216.84	716.92
Document	600.73	671.22	499.49	554.64	330.09
Requirement	993.12	1075.67	861.18		614.21

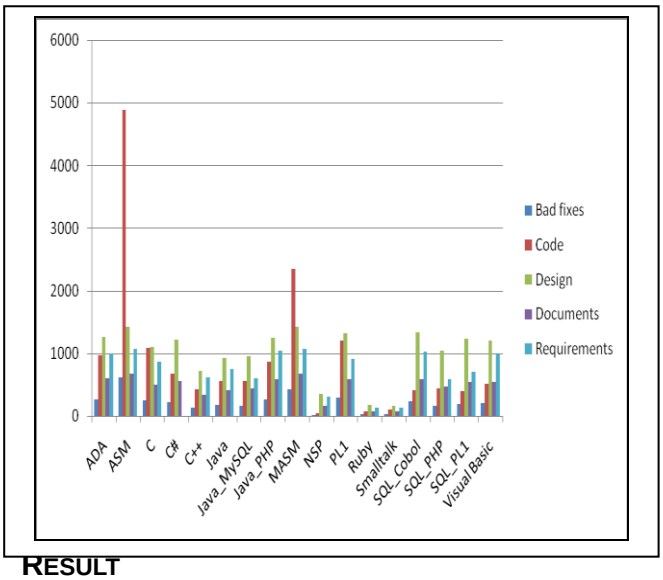
CONTINUED FROM ABOVE TABLE 1

Java	Java_MySQL	Java_PHP	MASM	NSP	PL1
177.75	166.17	256.92	419.95	20.93	299.20
551.45	550.15	868.68	2354.16	46.33	1202.82
922.06	950.26	1248.83	1418.83	350.30	1316.59
416.94	443.14	591.82	669.28	156.06	589.65
750.15	602.09	1038.75	1070.42	301.04	908.72

CONTINUED FROM ABOVE TABLE 1

Ruby	Smalltalk	SQL_Cobol	SQL_PHP	SQL_PL1	Visual Basic
29.24	27.89	231.29	162.56	184.71	206.95
71.13	97.35	404.79	433.19	400.74	507.43
175.81	156.61	1331.50	1038.84	1228.99	1204.54
79.39	71.32	591.50	463.58	548.26	537.71
137.83	134.59	1023.84	593.61	709.45	983.29

TABLE 3
FINDING RELATIONSHIP BETWEEN DEFECT AND LANGUAGES



4 RESULT

The graph shows that there are maximum error during ASM ie Assembly languages, MASM, PL1 Programing languages), The lowest error are in RUBY, SMALL TALK. The various reasons explained for these are mostly likely as below:

a) **ASM –** Assembly computer language – has syntax has defect that makes coding prone to error.

b) **MASM –** Microsoft Assembler - MASM syntax has some Significant defects that makes coding prone to error. Many of these Statement must be on a single line, max 128 chars

c) **PL1 -** Programming language 1 is a procedural, imperative computer programming language designed for scientific, engineering, business and systems programming application. It has been used by various academic, commercial and industrial organizations since it was introduced in the 1960s, and continues to be actively used.PL/I's principal domains are data processing, numerical computation, scientific computing, and systems programming; it supports recursion, structuredprogramming, linked data structure handling, fixed-point, floating-point, complex, character string handling and bitstring handling.

d) **Ruby -** Ruby is a dynamic, reflective, object-oriented, general-purpose programming language. It was designed and developed in the mid-1990s by Yukihiro "Matz" Matsumoto in Japan. According to its authors, Ruby was influenced by Perl, Smalltalk, Eiffel, Ada, and Lisp. Not as fast as Java dynamic type languages (like Python, Ruby) is less error prone.

e) **Small Talk –** object-oriented languages

f) C - is hard to beat,

g) Java – is verbose (i.e. it takes more code to get something done) reliable and expensive.

Functional and scripting languages tend to provide the most concise code, whereas procedural and object-oriented languages are significantly more verbose.

5 CONCLUSION

The object oriented languages like java, C# are less prone to error or defect as compared to other languages like lower level languages and middle level languages due robust, verbose nature of them. They have stricter rules to help prevent programming mistakes.

REFERENCES

[1] B.Kahanwal, " Abstraction level taxonomy of Programming languages framework ", *IJPLA*, Vol.3, No.4, Oct 2013.

[2] M. S. Rawat, S. K. Dubey, "Software Defect Prediction Models for Quality Improvement: A Literature Study," *IJCSI*, vol. 9, no. 5, pp. 288–296, 2012.

[3] G.Phipps, Comparing observed bug and productivity rates for Java and C++, Software Practice & Experience, Vol 29 Issue 4, Pages 345-358, April 10, 1999.

[4] D. K.Verma, H. S. Shukla, "Importance and Role of Programming Languages in Software Quality Improvement," *JARCSSE*, vol. 5, no. 11, pp. 205–209, 2015.

[5] P.Bhattacharya, I.Neamtiu"Assessing Programming Language Impact on Development and Maintenance: A Study on C and C++", *ICSE* '11, May 21–28, 2011.

[6] D. Renu, E. Ritika, "A Proposed Defect Tracking System for Classifying the Causes of Software Errors," *IJARCSSE*, vol. 5, no. 5, pp. 1150–1154, 2015.