CU Global Management Review
Volume 5, Issue 2, July-December 2016, pp. 86-96

SELECTION OF PROGRAMMING LANGUAGE FOR SOFTWARE DEVELOPMENT – CONJOINT APPROACH

- **Atul Shiva**, Assistant Professor in Commerce, Department of Commerce, Sri Aurobindo College of Commerce and Management, Ludhiana, Punjab, India
- **Vijay Chhabra**, Assistant Professor in Computer Sciences, Department of Computer Science, Sri Aurobindo College of Commerce and Management, Ludhiana, Punjab, India

Abstract

Purpose - Thepurpose of this research paper is to investigate the relationship between various attributes of programming languages for the development of software and selection of a best choice among them by the software developers.

Design/Methodology/Approach -For this purpose, five prominent programming languages were examined i.e. C, C++,Java, VB.NETand C#. The Conjoint Model was adopted and tested by multiple regression analysis for the formation of best choice in selected programming languages. Data was collected from students'pursuing post-graduate programs in various Colleges and Universities of Punjab State.

Findings -The main results of this empirical research suggest that VB.NET is the most acceptable programming language preferred for development of software by the participants followed by JAVA. In addition to this, participants prefer Platform Independence and Reliability factors much more than other factors.

Research limitations/implications -The paper provides comprehensive empirical evidence about the selection of a particular programming language by the participants and thus fills an important gap in Information Technology by using a statistical tool from academics, which can be used by corporate world while managing these programming languages. The major limitation is small sample size of 200 covering only one state of Punjab.

Originality/value–This paper is a modest attempt to develop and empirically test the validity and importance of computer programming languages through conjoint design.

Keywords: Computer Programs, Conjoint Analysis, Multiple Regression, Software Development, Software, Programming Language Preference.

JEL Codes: C880, M310& M390.

Introduction

Historically, the selection of the programming language for the development of software has been a process consisting of faculty evaluation, discussion and consensus. As the number of faculty, students and language option grows, this process is likely to become increasingly unwieldy (Parker and Chao, 2006). It has been argued that studying the history of the programming languages is essential as it helps developers avoid previously-committed mistakes in the development of the new languages (Wilson and Clark, 2000). It was also pointed out that an unfortunate trend in computer science is creating new language features without carefully studying previous work in this field (Grongo, 1999). Most of the books and article on the history of the programming languages tend to discuss languages in terms of generations where languages were classified by age (Cook, 1999). However the language adoption process has not been studied based upon the end user requirements, which are useful for the programming language developers. Understanding this process is a foundational step towards enabling language developers and advocates influencing its outcome and overall language use. Similarly, understanding will help developers in determining when and whether to bet on a new or experimental language or not (Meyerovich and Rabkin, 2013). Studies indicate that maintenance costs are at least 50% and sometimes more than 90% of the total cost associated with the software product (Koskinen, 2003, Seacord et al, 2003). If the Software developer wants to develop excellent software then he has to concentrate on user requirement, programming language, database management system, which places a pivotal role in the software development process (Joshi et al, 2012). The present paper introduces some guidelines for comparison and selection of appropriate programming language. It enables developers to base their decision in choosing the language on solid criteria that determines the suitability of the language to the application (Ghamdi and Urban, 1993). With the continuing increase in the variety, functionality and complexity in the engineering software along with its widespread use and increasing importance, more attention must be paid to programming language suitability, so that rational decisions regarding language selection may be made (Garcia et al, 2003). Thus, the plan of this study is to consider various preferences of end users and enable the programming language developers as to what current acceptance level of a language is and what best is required in order to remove the shortcomings at the language development stage. The selection criteria takes into account the programming feature of each language, the appropriateness of each of these features, the present and future industry acceptance of each language, the cost, reliability factors and the impact on the decision on the tactical and strategic direction of programming language developments by programmeror developers.

Review of Literature

Extensive work and research have been done in the field of comparative studies of the programming languages. Halang and Stoyenko (1990) conducted a comparative evaluation of seven high level real time programming languages and found that none of the languages actually used in industry is genuinely real-time. The study suggested some proposals to advance the inadequate state of affairs and ignited the discussion of this topic in real-time community. Garcia et al., (2003) reviewed C++, Java, LISP and Perl under the parameters of reusability, portability, reliability, readability, efficiency, availability of compilers and tools, familiarity and expressiveness. Aldrawiesh et al., (2009) presented review of the capabilities of the programming languages like Java, C++ and C# for developing web services and distributed systems based upon the criteria of simplicity and usage, concurrency, platform, maintainability, high integrity and reliability. Sleiman et al., (2010) reviewed ten programming languages with the aim of determining as to which programming language is more suitable for a particular language domain; Web Application Development and Object Oriented Based Abstraction. The Work provides evidence and analysis on why some languages are better than others or have greater advantages than others. However, Fajuyigbe et al., (2009) proposed an appropriate language for introducing Object Oriented Programming in tertiary institutions which helped the students to adopt a skill for using the most appropriate language. The study defined a set of criteria critical to the selection of most suitable programming language for introducing students to Object Oriented concepts and programming. The programming languages selected in this study were Java, C#, C++, C and Visual Basic. Joshi et al., (2012) discussed important factors for developers to select the programming languages as well as database management system. The study expressed that there is no thumb rule at all while selecting a particular programming language or DBMS. Jindal et al., (2013) studied about programming languages like C and C++ and C Objective and explained that one language effects and influences other language in one way or the other. Parker and Ottaway (2006) developed and documented an exhaustive set of selection criteria to the process of programming languages. The present study suggests comprehensive set of criteria while selecting a programming language like cost, life cycle of language, academic and industry dependence, ease of use, support for target application, object oriented support and others, which is helpful for teaching environment and proposes several approaches for the application of these criteria. Obi and Akwukuma (2013) made an assessment of programming language reliability utilizing soft-computing with reference to structured languages like C and C++ or Object oriented languages like Java, C# and Ruby. The paper determines reliability in terms of 'reliable', 'moderate reliable' and 'not reliable'. Based upon

these existing studies,this study conducted on major parameters of selectingprogramming languages. For thisfive prominent programming languages C, C++, Java, VB.NET and C# were examined. The major factors used for comparison are platform independence, security, development cost, reliability and ease of use.

Objectives of the Study

For studying the superiority of a particular social networking site, following objectives are considered in this study:

1. To analyze the impact of various factors affecting the choice of a particular programming language for software development by the IT companies.
2. To identify the best choice of a particular programming language among IT professionals (with reference to C, C++, Java, VB.NET and C#).

Hypothesis of the study

Ho1: There is no association between the dependent variable and the different explanatory variables.

Ho2: There is no absolute selection of a particular programming language by the IT professionals over other languages.

Research Methodology

For the purpose of estimating the research models for hypotheses test, firstly a sample of 200 students from Colleges and Universities offering advanced computer programming courses were selected, which covered the state of Punjab. To analyze and find out the best choice of a social networking site, Conjoint Analysis technique was used. Conjoint Analysis was first introduced into the marketing literature by Green and Rao (1971) and Johnson (1974). Green and Srinivasan (1978) indicated that conjoint analysis is an analytic method with de-compositional approach for evaluating the preference structure of the known assesses and overall assessment. Louviere and Islam (2008) described three methodologies of Conjoint Analysis:

- Traditional Conjoint Analysis, based on giving preference,
- Choice– based or Discrete Choice Conjoint Analysis, based on choice,
- Best/Worst (BW) Conjoint.

Out of these, Conjoint Analysis was used, where respondents were asked to rate or rank the product scenarios and the analysis reveals the relative importance, called utilities of each of the different levels of each attribute. Further,in the study the preference functions are measured by

ordinary least squares (OLS) regression method with dummy variables, since researches have shown that the efficiency of this technique is often similar to more complex techniques like Logit, Monanova, Linmap and others, but the results are easier to interpret (Oppewal, Vriens, 2000). The orthogonal arrays (Orthoplan) were generated by SPSS – 18.0. A total of 25 design cards were generated for the respondents to give preferences using 10 – point Likert Scale (1 = Least Preferred, 10 = Most Preferred). In our study, the number of stimuli was 25 which were higher than the minimum number of stimuli (total number of levels across all attribute – number of attributes + 1 = 20) that was evaluated by the respondents to ensure the reliability of the estimated parameters.

Research Design

The research study adopted content analysis in order to find the impact of various attributes like platform independence, security, development cost, reliability and ease of use to assess various programming languages like C, C++, JAVA, VB.NET and C# among the target audience. Table number 1 represents the orthogonal matrix in 25 cards as designed by SPSS 18.0 software. The non-probability sampling technique is used to collect the opinions from the respondents falling in the age group of 17 years to 25 years. The sample size is 200. The Conjoint Design framed on the above mentioned attributes are mentioned in Table number 1.

Table 1. Conjoint Design: Explanation of Attributes and Levels

ATTRIBUTES					
Programming Languages	**Platform Independence**	**Security**	**Development Cost**	**Reliability**	**Ease of Use**
ALTERNATIVES					
C	Full Support	Fully Secured	Very High Cost	Highly Reliable	Highly Difficult to Use
C++	Partial Support	Partially Secured	High Cost	Moderately Reliable	Difficult to Use
JAVA	No Support	Moderately Secured	Low Cost	Low Reliable	Easy to Use
VB.NET	Additional Component Support	Unsecured	Nominal Cost	Unreliable	Very Easy to Use
C#					

Total Number of Attributes=6
Total Number of alternatives across all attributes=25
Minimum Required Stimuli=25-6+1=20
Source: ***Authors' findings***

Analysis of Conjoint Results and Hypothesis Testing

Table number 3 represents the results of regression analysis, where the Adjusted R – Square value shows 89.7% variation in the model which is highly significant at 1 % level of significance, thus Null HypothesisHo1is rejected which signifies that there is association between the dependent variable and different explanatory variables. The value of Durbin – Watson statistic is 2.338, which lies in the range (1.25 – 2.75) showing that auto – correlation is not present.

Table 2: Model Summary

*Model Significant at 1% Level of Significance

Model Summary

R	R^2	Adjusted R^2	Std. Error of the Estimate	Change Statistics					Durbin-Watson
				R^2 Change	F Change	df 1	df 2	Sig. F Change	
.989[a]	.979	**.897**	.50990	.979	11.996	19	5	**.006**	2.338

a. Predictors: (Constant), VeryEasyto Use, Unreliable, Nominal Cost, Unsecured, Additional ComponentSupportRequired, C SHARP, Moderately Secured, C++, Easy to Use, Low Reliability, Low Cost, NoSupport, PartiallySecured, VB.NET, DifficulttoUse, Moderately Reliable, High Cost, PartialSupport, JAVA

b. Dependent Variable: OverallRating

Source: ***Authors' findings***

Table 3: ANOVA

ANOVA

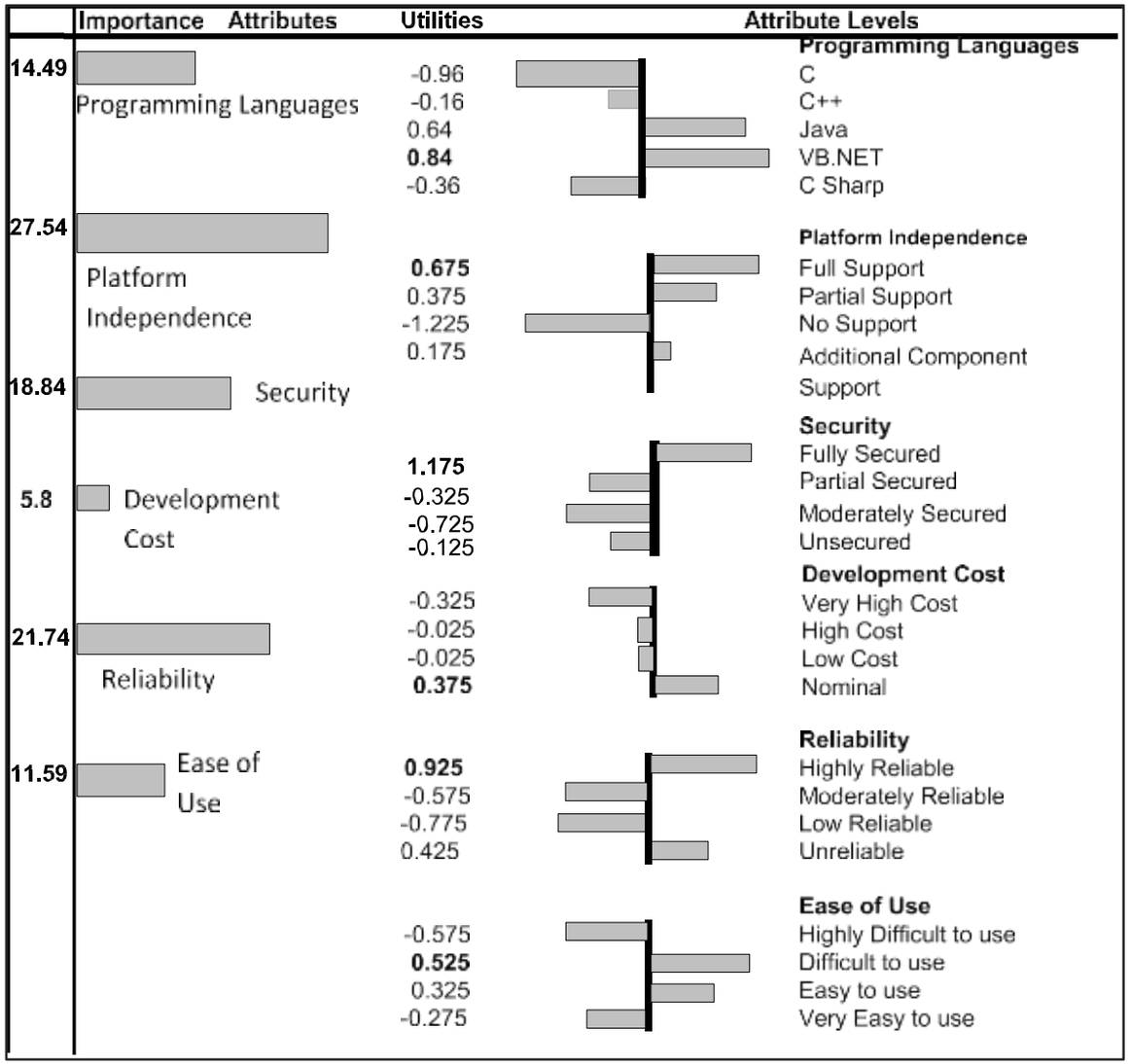
Model		Sum of Squares	Degree of Freedom	Mean Square	F	Sig.
1	Regression	59.260	19	3.119	11.996	**.006[a]**
	Residual	1.300	5	.260		
	Total	60.560	24			

b. Dependent Variable: Overall Rating

*Model significant at 1% Level of Significance.

Source: Authors' findings

Figure 1: Relative Importance and Utility Scores of Considered Attributes in the selection of a Programming Language



Source: Authors' findings

Figure 1 represents the mean preference structure of programming languages in terms of imporatance and utilities of various arttribute levels. All these help us to understand the rankings of attributes when it comes to the choice of a programming language for the development a particular software. In Figure 1, six salient attributes and their levels were identified for programming language choice process. Full Profile Conjoint Anlaysis was used for construction of preference structure. Analysing the preference strutcture or the importance accorded by the students to the six salient attributes, the students accorded the maximum utility/importance to

the attribute *'Platform Independence'* i.e Full Support with 27.54 percent importance. Here, the study concludes that the software developers highly require full support while developing any sort of software, thereby rejecting all other options of partial, no support and additional component required. However, software developers can agree to partial support with 0.375 utility in case full support option is not available. The second attribute adjudged best by software developers was '*Reliability Factors*' with an importance value of 21.74 percent in which the requirement is for highly reliable software with a utility value of 0.925. The reliabilty aspect is so important for the developers that for the other options they have given negative utilities.

Thereafter, at the third place in the worth hierarchy is the attribute of *'Securiy Aspects'* with importance of 18.84 percent. Here, it was observed that only highly reliabile software are generated and selected by the software developers with a utility values of 1.175. The mandate of repsondents was clear enough that in security no compromise could be made when they are put to use by the end users in the market. Then, at fourth place of the hierarchical framework, is the attribute *'Programming Languages'* with 14.49 percent importance. This is an important finding, as it shows that, our respondetns are more keen towards platform independence, secutiry and reliablity factors than type of language since there are many other langugaes which can be tested and used to produce a software for end users but the type is not the real important thing which is used while developing a software. VB.NET is the most preferred language by the respondents with 0.84 utilty followed by JAVA with 0.64 utility, while all other languages are not commonly preferred since they have been rated in negative utitlties in the study. The next preference is given to the *'Ease of Use'* attribute with an importance of 11.59% where the repsondents prefer that the final usage should not be highly difficlut but still it must require some training and experience to use the software so developed by the software professionls. The last attribute was the presence of *'Development Costs'* in the Conjoint Design with an importance of 5.8 percent only. The respondents have very less interest in the development cost since the software is designed after taking into consideration the demand, paying capacities and facilities required by the consumers. The software is customised according to the above mentioned parameters, therefore the cost factors are always considered last by the IT professionals.

In Table 4, the Best Choice of a Programming Language for the development of software (Final Result) is projected where the respondents have expressed their best choice using *"Fully Support Platform Independence – High Reliability – Fully Secured – VB.NET Language – Difficult to Use – Development Cost viabilities"*.

Table 4: Best Choice of Programming Language for Software Development (Final Result)

Best Choice	Full Support	Highly Reliable	Fully Secured	VB.NET	Difficult to Use	Nominal Cost
Utilities	0.675	0.925	1.175	0.84	0.525	0.375
Code	B1	E1	C1	A4	F2	D4

Source: Authors' findings

This best choice selection by the respondents by Conjoint Analysis clearly rejects the second Null Hypothesis Ho2 and leads to acceptance to alternative Hypothesis that there is one absolute selection of a particular programming language by the respondents over other languages based on various attributes and attribute levels.

Limitations of the Study

Our study is subject to some limitations too. First, a large part of our respondents were students and the sample size is 200. Taking into account the fact that the demographics of respondents and their requirements are constantly changing, further research should validate our findings with other population groups. In addition, the respondents in our sample were from a particular State i.e. Punjab, which limits the scope of our study. More studies covering zones like North India could further validate the findings of the present study. Finally, the results of every conjoint analysis are highly dependent on the choice of the attributes and their respective levels. Addressing this argument,the present study lays stress on decisions which were based on the extensive literature review combined with pre-study interviews and careful pretesting of the conjoint design.

Discussion and Concluding Remarks

Our study is an attempt to empirically investigate the factors behind the choice of a Programming Language for the development of software using a conjoint approach. The major findings of the study are that Platform Independence, Reliability and Security factors should be considered as the most important parameters for the development of software. The type of a programming language comes afterward with VB.NET and JAVA as preferred options followed by usage and development costs factors considered by the respondents. This research paper provides comprehensive empirical evidence about the selection of a particular programming language by the participants and thus fills an important gap in Information Technology research by using a statistical tool from Academics.The findings of the research study can be used by corporate world while managing these programming languages.

References

Aldrawiesh, K. Al-Ajlan, A. Al-Saawy Y.and Bajahzar, A., (2009). A Comparative Study between Computer Programming Languages for Developing Distributed Systems in Web Environment. *Software Technology Research Laboratory (STRL) De Montfort University*, Leicester, UK.

Cook, D. (1999). Evolution of Programming Languages and why a Language is not enough to solve our problems. *Software Technology Support Center*, available online (Last accessed: November 2010) http://www.stsc.hill.af.mil/crosstalk/1999/12/cook.asp

Fajuyigbe O., Onibere E., Ekuobase G. (2011), Appropriate language for introducing object oriented programming in tertiary institutions retrieved online from www.ajol.info/index.php/ijonas/article/view/49925

Garcia, R. Jarvi, A. Lumsdaine, A. Siek, J. and J. Willcock (2003). A Comparative Study of Language Support for Generic Programming. *Open Systems Lab*, Indiana University Bloomington, IA.

Ghamdi, A. J. and Urban, J. (1993). Comparing and Assessing Programming Languages: Basis of a Qualitative Methodology. *Symposium on Applied Computing*. SAC'93 Proceedings of the 1993 ACM.SIGAPP symposium on Applied Computing: states of the art and practice. pp. 222-229.

Green, P. E. and Rao, V. R. (1971). Conjoint Measurement for Quantifying Judgmental Data, *Journal of Marketing Research,* Vol. 8, pp. 355 – 363.

Green, P. E. and Srinivasan, V. (1978). Conjoint Analysis in Consumer Research: Issues and Outlook, *Journal of Consumer Research,* Vol. 5, pp. 103 – 123.

Grogono, P. (1999). The Evolution of Programming Languages, *Course Notes, Department of Computer Science*. Concordia University, Montreal, Quebec, Canada.

Halang, W. A. and Stoyenko, A. D. (1990). Comparative Evaluation of High-Level Real-Time Programming Languages. *Real-Time Systems,* Vol (2), pp. 365-382.

Jindal, G. Khurana, P. and Goel, T. (2013). Comparative study of C, C Objective, C++ programming language. *International Journal of Engineering and Computer Science.* Vol (2) Issue 1, pp. 202-206.

Joshi, O. D. Gundale, V. A. Jagdale, S. M. (2012). Guidelines in Selecting a Programming Language and a Database Management System. *International Journal of Advances in Engineering & Technology,* Vol. 3, Issue 1, pp. 137-144.

Johnson, R. M. (1974). Trade – off Analysis of Consumer Values, *Journal of Marketing Research.* 11 May, pp. 127–127.

Koskinen, J. Software maintenance cost,Sept 2003. http://users.jyu.fi/~koskinen/smcosts.htm.

Louviere, J. and Islam, T. (2008). A Comparison of importance weights/measures derived from choice – based conjoint, constant sum scales and best-worst scaling, *Journal of Business Research,* Vol 61(9), pp. 903-911.

Meyerovich, L. A. and Rabkin, A., (2013). Empirical Analysis of Programming Language Adoption. *Oopsla*. Available online (Last accessed: June 2014) http://sns.cs.princeton.edu/docs/asr-oopsla13.pdf

Mouaaz, N. and Adi, M. (2012). Choosing Appropriate Programming Language to Implement Software for Real-Time Resource-Constrained Embedded Systems-Theory and Design Methodology, Dr. Kiyofumi Tanaka (Ed.), ISBN: 978-953-51-0167-3, InTech, Available from :*http://intechopen.com/books/embedded-systems-theory-and-design-methodology/choosing-appropriate-programming-language-to-implement-software-for-real-resource-constrained-e*

Obi, J. C. and Akwukuma, V.V.N. (2013). Assessment of Programming Language Reliability Utilising Soft-Computing. *International Journal on Computational Sciences & Applications,* Vol. 3(3), pp. 25-35.

Oppewal, H. and Vriens, M. (2000). Measuring perceived service quality using conjoint experiments, *International Journal of Bank Marketing,* Vol. 18, No 4, pp. 154-169.

Parker, K. R. Ottaway, T. A. and Chao, J. T., (2006). Criteria for the Selection of a Programming Language for Introductory Courses, *Int. J. Knowledge and Learning,* Vol. 2, Nos. ½, pp. 119-139.

Seacord, R. C.Plakosh,D. and Lewis, G. A. (2003). ModernizingLegacy Systems. Software Technologies, *Engineering Process and Business Practices*. Addison-Wesley.

Sleiman R., Jiang Li., Mingzhi L., Yuanwei L. (2010), Comparative Studies of 10 Programming Languages within 10 Diverse Criteria retrieved from *http://arxiv.org/abs/1009.0305*

Wilson, L.B. and Clark, R.G. (2000). Comparative Programming Languages, *Addision-Wesley.*