# APPLICATION OF TAGUCHI'S PARAMETER DESIGN FOR WEBSITE COMPATIBILITY TESTING

**M Shilpa[1], T Shivakumar[2]**

*[1]Dept. of Industrial Engineering and Management, M S Ramaiah Institute of Technology, Bangalore, India– 560054*
*[2]Aditi Technologies Pvt. Ltd.,Manyata Tech Park, Nagavara, Bangalore, India– 560064.*

## Abstract

*Testing the product just before its release to the customer is very important as it marks the success of that product. In software development, testing of the software is conducted at the last stage of the development cycle. During software testing, there will a large number of variables and their possible values. For this if conventional testing process becomes cumbersome and hence a more efficient testing strategy has to be applied. This paper discusses the application of Orthogonal Array based Software Testing (OATS) for minimizing the number of bugs in the software. In this paper, Taguchi's Orthogonal Array (OA) is applied for compatibility testing of a website, by considering five control factors and one interaction between the factors. Experiments have been conducted using L18 orthogonal array; experimental results are analyzed using Smaller-The-Better (STB) type of Signal-to-Noise (S/N) ratio and optimal levels for the control factors are obtained. With these optimal levels it is found that the number of bugs has been considerably reduced.*

***Keywords:*** *Orthogonal arrays, Smaller-the-better, Signal-to-Noise ratio, Website compatibility testing*

---------------------------------------------------------------------***---------------------------------------------------------------------

## 1. INTRODUCTION

Testing of a product is checking and evaluating whether the entire system satisfies the requirements specified by the customer. Not only that, it also helps to identify the areas of improvement vis-à-vis customer requirements. According to ANSI/IEEE 1059 standard, software testing is defined as 'a process of analyzing a software item to detect the differences between existing and required conditions (that is defects/errors/bugs) and to evaluate the features of the software item'.

Software testing has now become a very complex and challenging task. [1]To meet this challenge, it is necessary to build right strategy that provides a road map for testing. Software testing is a sub set of Software Quality Control (QC). Software QC ensures the verification of developed software with respect to the documented requirements. Software Quality Assurance (QA) is different from software QC in a way that it ensures the implementation of processes, procedures and standards in context to verification of the developed software and intended requirements.

### Advantages of Software Testing:

There will be a significant cost reduction and rework time will also be reduced if the testing is started early [1]. However in Software Development Life Cycle (SDLC) one can start testing the software from the phase in which customer requirements are gathered. It can continue until the deployment stage of the software.

There are many advantages of testing the software; following are a few of them.

- Huge savings in terms of time and money if the software defects are identified earlier.
- The development downtime can be avoided
- If the software application is excellent then one can provide better customer service
- One can identify improvements for later versions
- Certain modules could be identified and catalogued for reuse at a later point of time
- Sufficient training needs can be identified for programmers and developers

### Types of Software Testing:

There are two types of software testing depending on the role of the end user. [1]The first one is the manual testing wherein the software is tested manually. Here any type of automated tool or script will not be made use of. Here, the tester takes over the role of an end user and tests the software to identify any un-expected behaviour or bug. The test plan, test cases or test scenarios will be used by the test engineers use to test the software for completeness of testing.

The other type of testing is automation testing. This type of testing is also called as Test Automation. Here, the test engineer writes scripts and tests the application using software. The advantage of this type of testing is that the test scenarios that are performed manually can be rechecked. This is shown in figure 1. Also, automation testing has the benefit of a wider test coverage and accuracy, thus helps in saving time and resources when compared to manual testing.

Levels of Software Testing:

Levels of testing envisage the various methodologies used during software testing. [1]The two important levels of software testing:

- Functional Testing.
- Non-Functional Testing.

Functional Testing belongs to the category of black box testing wherein the application is tested with respect to the specifications. The application testing happens by changing the input and then checking the output for conformance to the functionality which is intended for.

Testing the application for attributes like performance, security, user interface etc. do not affect the function but play a significant role to the user, is called as non-functional testing.

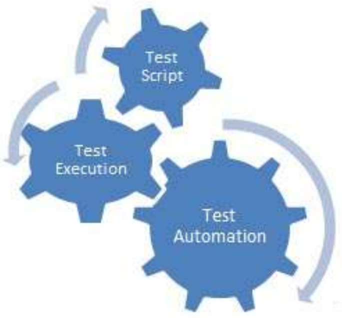


**Fig 1:** Automation Testing[1]

## 2. SOFTWARE TESTING PROCESS FLOW

Software testing has its own life cycle starting from preparing test strategy to documentation. All through the SDLC, it is essential to conduct useful and constructive testing, leading to reduced errors. Figure 2 shows the software testing process flow.

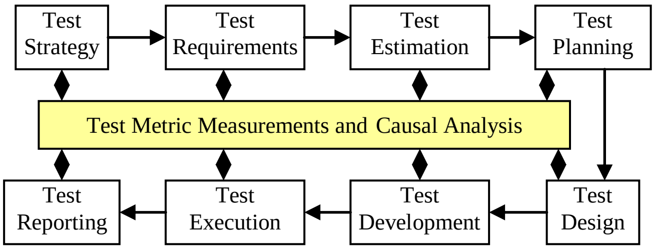


**Fig 2:** Software Testing Process Flow[9]

Any practice adapted by the test engineer should be effective and also utilize minimum resources for testing, saving time and money.

## 3. TAGUCHI'S ORTHOGONAL ARRAY APPROACH

Dr. Genichi Taguchi has revolutionized the very idea of quality control by the introduction of robust design concepts. [3]He has envisaged a new method of designing experiments called as 'Orthogonal Array' (OA). These standard arrays specify the way of carrying out the nominal number of experiments by providing the complete information of all the factors that influence the performance of the product. [4]The core of the OA lies in selecting the optimal level combinations for the input variables, thus making the product insensitive to the environmental conditions. [5]The OA has the following special properties that reduce the number of experiments to be conducted.

i) Every column of an OA has a special combination of level settings, all the levels appearing equal number of times, which is called the balancing property.

ii) All the level settings of the factor are used for conducting the experiments

iii) The sequence of level settings should not be changed while conducting the experiment.

### 3.1 Orthogonal Array Testing Strategy

Taguchi's OA approach has been successfully applied in many of the manufacturing and electronics engineering applications.

Adapting Taguchi's OA for software testing throughout the SDLC leads to test effectiveness with reduced number of test cases[6]. Due to the use of orthogonal arrays, the number of test cases required for testing the website compatibility reduces and hence results in huge savings in cost. The experimental results remain the same[7].

### 3.2 Compatibility Testing

It is a non-functional testing which ensures that application/website/system is capable of running on various objects like various browsers, various resolutions, various operating systems, various networks and with some other applications [8].

This paper discusses the application of OATS for compatibility testing of a website, which has been used in a particular project to minimize the number of bugs. For the identification of control factors, a brainstorming session has been conducted with the software development team, testing team, project managers and end users. Although there are many control factors, significant factors alone are considered for the experimentation. The levels of these factors are considered in such a way they ensure coverage of the entire range. These levels have been coded for data confidentiality purpose. From the brainstorming session, it was strongly felt that interaction between Network and Operating Systems is significant; hence this interaction is also considered for experimentation. The control factors, their levels and interaction are presented in table 1. During software testing, the website is checked for its compatibility with different web browsers, operating systems, varying parameters of network such as bandwidth, operating speed etc., enhanced versions of the software and mobile platforms.

**Table 1:** Control Factors and Their Levels

| **Control Factors** | **Symbols** | **Level 1** | **Level 2** | **Level 3** |
|---|---|---|---|---|
| Web Browser (WB) | A | WB1 | WB2 | WB3 |
| Operating | B | OS1 | OS2 | OS3 |

| | | | | |
|---|---|---|---|---|
| System (OS) | | | | |
| Network (N) | C | N1 | N2 | N3 |
| Software Version (SV) | D | SV1 | SV2 | - |
| Mobile Platform (MP) | E | MP1 | MP2 | - |
| N * OS | A * B | - | - | - |

**Selection Of Appropriate Orthogonal Array For Mixed Levels Design Involving Interaction**

With three factors at three levels, two factors at two levels and one interaction, using full factorial approach, the number of experiments is $3^3 * 2^2$ (for factors) + 4 (for interactions) = 112 i.e. the number of experiments is very high, which is time consuming and expensive. With the use of Taguchi's OA approach, the number of experiments required can be reduced and the results of full factorial and OA approaches are almost same. Hence, OA approach is selected for experimentation. [5]The calculations for minimum number of experiments required are shown below.

Degree of freedom (dof) = No. of factors * (no. of levels -1)
With three control factors at three levels each
Dof = 3 factors * (3 -1)
Dof = 6
With two control factors at two levels each
Dof = 2 factors * (2 -1)
Dof = 2

The dof calculations for interactions are shown below.
AxB
Dof = (3-1) * (3-1) = 4
The minimum number of experiments = 6+2+4+1(overall mean) = 13 experiments

Thus, an OA with a minimum of thirteen rows and six columns are required for the experimentation, of which, at least three columns should have three levels. Hence, $L_{18}$ OA containing eighteen rows, and seven columns with three levels and one column with 2 levels, is found appropriate and is selected for experimentation. The two-level factors 'D' and 'E' have to be allocated to two columns having two levels.

Using the dummy level technique, factor E having two levels is assigned to sixth column of the OA, which has three levels. Wherever level 3 is present in this column, factor E is kept at level 2 to check the compatibility with a higher software version. Therefore, Dummy level $E_2$ is maintained wherever level three is present in sixth column[6].

The required linear graph, standard linear graph of $L_{18}$ OA and modified linear graph are shown in figures 3, 4 and 5 respectively. The $L_{18}$ OA along with the physical layout prepared for the experimentation is shown in table 2.

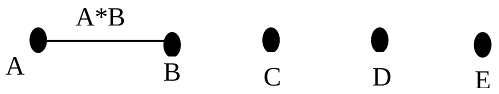


**Fig 3:** Required Linear Graph[2]

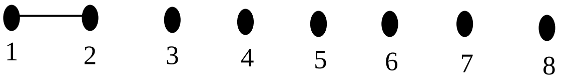
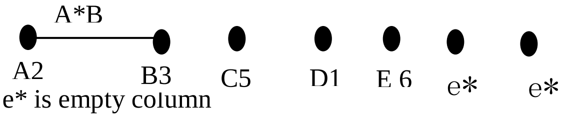


**Fig 4:** Standard Linear Graph of $L_{18}$ OA[2]



e* is empty column

**Fig 5:** Modified Linear Graph[2]

**Table 2:** Physical Layout of $L_{18}$ OA for Experimentation

| **Expt. No./ Symbols** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
|---|---|---|---|---|---|---|---|---|
| | **D** | **A** | **B** | **A*B** | **C** | **E** | **e*** | **e*** |
| 1 | SV1 | WB1 | OS1 | | N1 | MP1 | | |
| 2 | SV1 | WB1 | OS2 | | N2 | MP2 | | |
| 3 | SV1 | WB1 | OS3 | | N3 | MP2 | | |
| 4 | SV1 | WB2 | OS1 | | N2 | MP2 | | |
| 5 | SV1 | WB2 | OS2 | | N3 | MP2 | | |
| 6 | SV1 | WB2 | OS3 | | N1 | MP1 | | |
| 7 | SV1 | WB3 | OS1 | | N1 | MP2 | | |
| 8 | SV1 | WB3 | OS2 | | N2 | MP1 | | |
| 9 | SV1 | WB3 | OS3 | | N3 | MP2 | | |
| 10 | SV2 | WB1 | OS1 | | N3 | MP2 | | |
| 11 | SV2 | WB1 | OS2 | | N1 | MP2 | | |
| 12 | SV2 | WB1 | OS3 | | N2 | MP1 | | |
| 13 | SV2 | WB2 | OS1 | | N3 | MP1 | | |
| 14 | SV2 | WB2 | OS2 | | N1 | MP2 | | |
| 15 | SV2 | WB2 | OS3 | | N2 | MP2 | | |
| 16 | SV2 | WB3 | OS1 | | N2 | MP2 | | |
| 17 | SV2 | WB3 | OS2 | | N3 | MP1 | | |
| 18 | SV2 | WB3 | OS3 | | N1 | MP2 | | |

## 4. EXPERIMENTATION AND DATA COLLECTION

Eighteen experimental runs are conducted according to the physical layout. Each experimental run is conducted thrice (three replications); the number of bugs is noted for each replication and then average of the results of the three replications is computed. The experimental results are shown in table 3.

**Table 3:** Results of the Experimental Runs

| **Expt. No. / Symbols** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **Number of bugs detected** | | | **Average** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **D** | **A** | **B** | **A*B** | **C** | **E** | **e*** | **e*** | **1** | **2** | **3** | |
| **1** | **SV1** | **WB1** | **OS1** | | **N1** | **MP1** | | | **58** | **60** | **58** | **58.67** |
| **2** | **SV** | **WB** | **OS** | | **N** | **MP** | | | **65** | **68** | **66** | **66.33** |

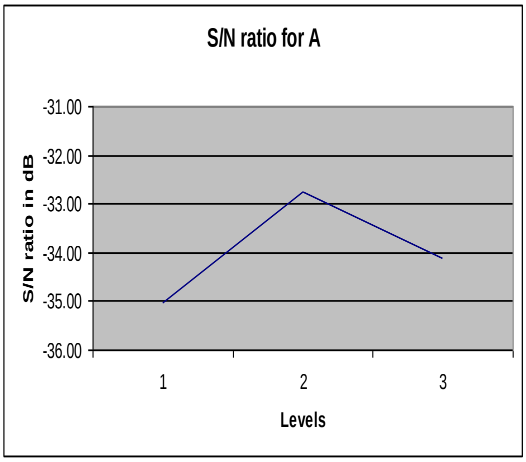
| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 1 | 2 | | 2 | 2 | | | | | | |
| 3 | SV 1 | WB 1 | OS 3 | | N 3 | MP 2 | | | 42 | 42 | 41 | 41.67 |
| 4 | SV 1 | WB 2 | OS 1 | | N 2 | MP 2 | | | 60 | 61 | 60 | 60.33 |
| 5 | SV 1 | WB 2 | OS 2 | | N 3 | MP 2 | | | 55 | 57 | 56 | 56.00 |
| 6 | SV 1 | WB 2 | OS 3 | | N 1 | MP 1 | | | 45 | 48 | 46 | 46.33 |
| 7 | SV 1 | WB 3 | OS 1 | | N 1 | MP 2 | | | 44 | 45 | 44 | 44.33 |
| 8 | SV 1 | WB 3 | OS 2 | | N 2 | MP 1 | | | 47 | 48 | 45 | 46.67 |
| 9 | SV 1 | WB 3 | OS 3 | | N 3 | MP 2 | | | 52 | 51 | 52 | 51.67 |
| 10 | SV 2 | WB 1 | OS 1 | | N 3 | MP 2 | | | 48 | 48 | 49 | 48.33 |
| 11 | SV 2 | WB 1 | OS 2 | | N 1 | MP 2 | | | 65 | 67 | 65 | 65.67 |
| 12 | SV 2 | WB 1 | OS 3 | | N 2 | MP 1 | | | 62 | 61 | 62 | 61.67 |
| 13 | SV 2 | WB 2 | OS 1 | | N 3 | MP 1 | | | 35 | 36 | 33 | 34.67 |
| 14 | SV 2 | WB 2 | OS 2 | | N 1 | MP 2 | | | 38 | 37 | 39 | 38.00 |
| 15 | SV 2 | WB 2 | OS 3 | | N 2 | MP 2 | | | 34 | 32 | 32 | 32.67 |
| 16 | SV 2 | WB 3 | OS 1 | | N 2 | MP 2 | | | 46 | 48 | 45 | 46.33 |
| 17 | SV 2 | WB 3 | OS 2 | | N 3 | MP 1 | | | 62 | 65 | 64 | 63.67 |
| 18 | SV 2 | WB 3 | OS 3 | | N 1 | MP 2 | | | 54 | 54 | 55 | 54.33 |

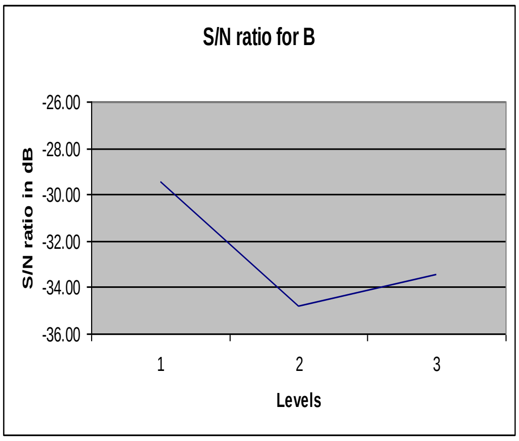
## 4.1 Computational Analysis

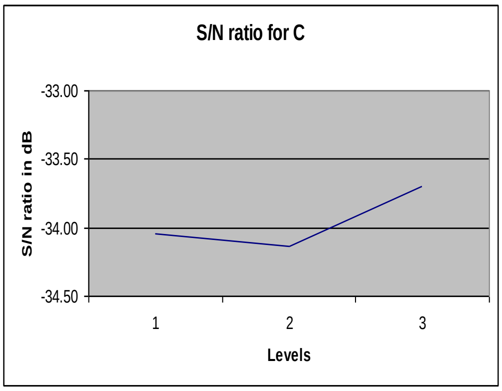
The ratio of Signal to noise takes both the mean and variability into account. In its simplest form, this ratio is the ratio of the mean (signal) to the standard deviation (noise). This ratio (η) for STB quality characteristic is given by the following equation.

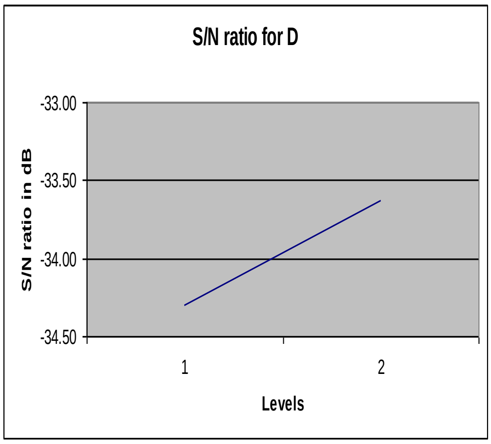
$$\eta = -\ 10 \log_{10} 1/n\ \Sigma\ y_i^2 \text{ decibels}$$

The experimental results have been analyzed using Taguchi’s STB type of S/N ratio and the same is presented in table 4. The S/N ratio is calculated for every factor level and is presented in table 5. The S/N ratio graphs are plotted for each of the factors and are shown in figure 6. From the S/N ratio graphs, the optimum levels for the four factors are identified and are presented in table 6.









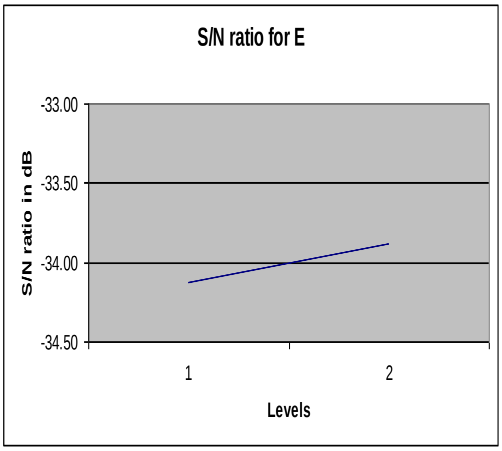


**Fig 6:** S/N Ratio Graphs for Every Factor

Confirmation experiment is conducted by placing the factors at their optimal levels[9]; compatibility testing has been carried out and number of bugs found is minimum.

**Table 4:** S/N Ratio for Number of Bugs

| Expt. No. / Symbols | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | S/N ratio (in dB) |
|---|---|---|---|---|---|---|---|---|---|
| | **D** | **A** | **B** | **A*B** | **C** | **E** | **e*** | **e*** | |
| 1 | SV1 | WB1 | OS1 | | N1 | MP1 | | | -35.37 |
| 2 | SV1 | WB1 | OS2 | | N2 | MP2 | | | -36.44 |
| 3 | SV1 | WB1 | OS3 | | N3 | MP2 | | | -32.40 |
| 4 | SV1 | WB2 | OS1 | | N2 | MP2 | | | -35.61 |
| 5 | SV1 | WB2 | OS2 | | N3 | MP2 | | | -34.96 |
| 6 | SV1 | WB2 | OS3 | | N1 | MP1 | | | -33.32 |
| 7 | SV1 | WB3 | OS1 | | N1 | MP2 | | | -32.94 |
| 8 | SV1 | WB3 | OS2 | | N2 | MP1 | | | -33.38 |
| 9 | SV1 | WB3 | OS3 | | N3 | MP2 | | | -34.26 |
| 10 | SV2 | WB1 | OS1 | | N3 | MP2 | | | -33.69 |
| 11 | SV2 | WB1 | OS2 | | N1 | MP2 | | | -36.35 |
| 12 | SV2 | WB1 | OS3 | | N2 | MP1 | | | -35.80 |
| 13 | SV2 | WB2 | OS1 | | N3 | MP1 | | | -30.80 |
| 14 | SV2 | WB2 | OS2 | | N1 | MP2 | | | -31.60 |
| 15 | SV2 | WB2 | OS3 | | N2 | MP2 | | | -30.29 |
| 16 | SV2 | WB3 | OS1 | | N2 | MP2 | | | -33.32 |
| 17 | SV2 | WB3 | OS2 | | N3 | MP1 | | | -36.08 |
| 18 | SV2 | WB3 | OS3 | | N1 | MP2 | | | -34.70 |

**Table 5:** S/N Ratio for Number of Bugs

| **Factors/ Levels** | **A** | **B** | **C** | **D** | **E** |
|---|---|---|---|---|---|
| **1** | -35.01 | -29.47 | -34.05 | -34.30 | -34.13 |
| **2** | -32.76 | -34.80 | -34.14 | -33.62 | -33.88 |
| **3** | -34.11 | -33.46 | -33.70 | - | - |

**Table 6:** Optimal Levels for the Control Factors

| **Symbol** | **Factor** | **Optimum Level** | **Value** |
|---|---|---|---|
| A | Web Browser (WB) | 2 | WB2 |
| B | Operating System (OS) | 1 | OS1 |
| C | Network (N) | 3 | N3 |
| D | Software Version (SV) | 2 | SV2 |
| E | Mobile Platform (MP) | 2 | MP2 |

## 5. CONCLUSION

From the analysis of results, it is evident that Taguchi's OA approach has led to the identification of optimal levels of the control factors that give desired value of the quality characteristic. This approach has been applied to reduce the number of bugs during compatibility testing of a webpage; this is STB type of quality characteristic. Five control factors at mixed levels along with an interaction are considered for experimentation and the experimentation has been appropriately conducted with $L_{18}$ OA. The S/N ratio is calculated for every factor level and the optimal levels for these control factors are identified. With these optimal levels, confirmation run has been carried out and the number of bugs found is minimum.

## REFERENCES

[1] Ljubomir Lazic and Nikos Mastorakis, Orthogonal Array application for optimal combination of software defect detection techniques choices, WSEAS TRANSACTIONS on COMPUTERS, 7(8):1319-1336, 2008.

[2] Naidu N V R and Dharani Gowda. Optimizing Process Parameters using Taguchi Techniques for Quality Improvement- A Case Study, Industrial Engineering Journal, 30(9):14 – 23, 2001.

[3] Palanikumar K. Cutting Parameters Optimization For Surface Roughness In Machining Of Composites Using Taguchi's Method, Journal of Reinforced Plastics and Composites, 2(16):1739-1751, 2006.

[4] Resit Unal and Edwin Dean. Taguchi Approach To Design Optimization For Quality And Cost: An Overview, Annual Conference of International Society of Parametric Analysts, 1991.

[5] Shilpa M and Naidu NVR. Quality Improvement in Multiresponse Experiments Through Robust Design Methodology, International Journal of Quality Research, 6 (2):137-141, 2012.

[6] Shilpa M and Naidu NVR. Quantitative Evaluation of Quality Loss For Fraction Defective Case Using Taguchi's Quality Loss Function, International Journal of Logistics Systems and Management (IJLSM) special Issue on: Modeling Supply Chain Planning Problems by Integrating Data Analysis and Optimization Techniques, 18(1):126- 138, 2014.

[7] Shubhra Banerji. Orthogonal Array Approach for Test Case Optimization, International Journal of Advanced Research in Computer and Communication Engineering,1(9):613 – 621, 2012.

[8] Sreenivasa Pisupati. Software Product Testing Using Orthogonal Array (OA) Testing Technique, International Journal of Technology Enhancements And Emerging Engineering Research, 3(8):181 – 186, 2015.

[9] Teruo Mori and Shih-Chung Tsai. Taguchi Methods: Benefits, Impacts, Mathematics, Statistics And Applications, American Society of Mechanical Engineers Press, 2011.