ISSN 2319–7757 EISSN 2319–7765

Indian Journal of Engineering

Software Information Flexibility for Lean Six sigma Software Development Using Multiple Regression Analysis

Chandrakanth G Pujari[1]

Department of Master of Computer Applications, Dr. Ambedkar Institute of Technology Bangalore, 560056, India; Email: varuncp@rediffmail.com

☼Corresponding author: Dr. Chandrakanth G Pujari Department of Master of Computer Applications, Dr. Ambedkar Institute of Technology Bangalore-560056, India; Email: varuncp@rediffmail.com

Publication History
Received: 5 January 2017
Accepted: 27 January 2017
Published: April-June 2017

Citation
Chandrakanth G Pujari. Software Information Flexibility for Lean Six sigma Software Development Using Multiple Regression Analysis. *Indian Journal of Engineering*, 2017, 14(36), 95-107

ABSTRACT

Software development organization is under threats to improve software productivity and reducing expenditure through minimization of wastage of software information's. This article presents a software development company to execute lean six sigma software project developments through information flexibility and also explain the various areas of future activity for improving lean six sigma software project developments. The work has been conducted using the flexible software system methodology framework. For computing information (software developer and software tools) flexibility and lean six sigma software project development, all variables achieving towards software developer's flexibility, software tools flexibility and lean six sigma software developments are identified. To determine their corresponding weights, fuzzy analytical hierarchy process (FAHP) has been employed. The present research includes only software developers and software tools as the elements of information flexibility. Other software information's may also be included to compute overall software information flexibility. The present research provides direction to assess the status of information flexibility and lean six sigma software developments. This research will help the software industries management to identify the problems to manage information flexibility and execute an effective lean six sigma software development. In this research, the theoretical perspective has been used and associated with software information flexibility and lean software development.

Keywords: Software information flexibility, Regression analysis, Software organization, lean six sigma software development

Abbreviations: FAHP - Fuzzy Analytical Hierarchy Process

1. INTRODUCTION

Software development systems in the modern days have become complexity as a variety of software's, software tools, software platforms and other inputs are being employed for software production purpose [1][2][3]. This complexity along with software market uncertainties require the software development system to respond quickly to changes [4][5]. Today, price of the software product is also set in the market by the software customers. So, reducing wastage of software information's and moving towards execution of lean six sigma software development have become important strategies to achieve price cutting [6][7][8]. Achieving higher levels of software productivity in this complex software environment requires a software system to rapidly adjust itself to complexities, uncertainties, and changes [9][11]. Software developments face an increasingly uncertain external environment as the rate of change in software customer expectations, global competition, and software technology accelerates [34]. Software researchers and software managers contend that flexibility is a strategic imperative that enables industries to cope with uncertainty [25]. Flexibility is the software company's ability to meet an increasing variety of software customer expectations without excessive prices, duration, software industries performance losses. Software flexibility is the ability to do things differently or do something else should the requirement arise [30] [32][33] reports that Flexibility is also the ability of a software system to respond or react to a change with minimum penalty in time, effort or price [34][35][36]. The change may be inside or outside. Lean six sigma software developments is a systematic approach for identifying and removing waste in operations through continuous development, minimizing operating price of the software system and fulfilling software customers' desire for more value at the lowest cost [37][38][39]. Software developer and software tools are among the important information of software Development Company which can significantly contribute to execute lean six sigma software project development shown by [40]. Some software managers have quoted software information flexibility (software developers and software tools) as the lowest level dimensions of software development flexibility and believe that these serve as the building blocks of the hierarchy. Software information flexibility in the form of software developer and software tools flexibility can be judiciously exploited towards minimization in wastages of software information's of software Development Company to achieve lean six sigma software development [30][32][42]. This paper presents to managing software developer and software tools flexibility to execute lean six sigma software development in industry.

2. METHODOLOGY

The flexibility of software system methodology framework is based on the situational philosophy and believed in software developing a unique approach for each problem deriving inspiration from the software developments in software system methodology in terms of schemes of software systems techniques using the programmed as well as creative approaches in an interactive behavior. The philosophy of combined methods is generalized over the whole spectrum of software system techniques in the framework of flexible software system methodology framework [42]. It is an evolving approach which can take care of the varied software requirements of problem situations in a flexible manner, for the present research, [26] from the basis for identification of software information flexibility variables and for lean six sigma software development variables. Ten variables contributing toward software developer flexibility, seven variables contributing toward software tools flexibility and nine variables contributing toward lean six sigma software development are considering for present research as shown in Tables 1-3, respectively. Weights of some variables are more than the others [22][21]. To determine their relative weights, FAHP has been employed [9][10][11]. Each variable has been compared with other variables pair-wise. Three experts; one software manager, one senior software production executive and academician are selected to cover the different fields and increase the software reliability of the FAHP. They, however, filled the response in qualitative score of very low, low, medium, high and very high as the difference between the importances of two variables. These quality measurement answers are converted in the quantitative values using the grades as: very low -1; low-3; medium-5; high-7 and very high-9. Their contributing weights to software developers' flexibility, software tools flexibility and lean six sigma software developments are calculated by drawing position matrix from the software experts' response. The weight of each variable has been determined by computing an Eigen vector and normalizing it for each software expert's answers. To obtain higher degree of accuracy, the final weightage of each variable of software developer flexibility, software tools flexibility and lean six sigma software developments are computed by taking mean of weightage computed for each software expert as depicted in Tables 1, 2, 3 respectively.

Table 1 Weightage of software developer flexibility variables

Sr. No.	Variables	Software Champion-I weight	Software Champion-II weight	Software Champion-III weight	Average weight	Standard deviation
1	*SDF1-Ability of software developers to develop on different software platform*	0.350	0.370	0.340	0.353	0.017
2	SDF2-Programming skill level of software developers to perform different projects	0.210	0.111	0.197	0.173	0.052
3	SDF3-Cost effectiveness of software developers over software project change	0.023	0.022	0.043	0.029	0.012
4	SDF4-Software reliability of software developers over software project change	0.112	0.111	0.098	0.107	0.008
5	SDF5-Attitude of software developers towards software project change	0.052	0.047	0.020	0.039	0.017
6	SDF6-Software productivity effectiveness due to change	0.052	0.111	0.043	0.068	0.037
7	*SDF7-Co-operation of software developers in achieving software production targets*	0.023	0.047	0.043	0.037	0.013
8	SDF8-Ability of software production developers to perform code inspection of software projects	0.052	0.047	0.098	0.065	0.028
9	SDF9-Ability of software production developers to do autonomous maintenance	0.112	0.111	0.098	0.107	0.008
10	SDF10-Training of software developers	0.023	0.022	0.020	0.021	0.012

Table 2 Weightage of software tools flexibility variables

Sr. No.	Variables	Software Champion-I weight	Software Champion-II weight	Software Champion-III weight	Average weight	Standard deviation
1	*STF1-Ability of software tools to perform diverse set of operations*	0.400	0.389	0.418	0.402	0.015
2	STF2-Software tools setup or change over	0.024	0.037	0.051	0.037	0.013
3	STF3-Time & effort needed to change the software tools and operations	0.101	0.085	0.051	0.079	0.025
4	STF4-Software cost effectiveness of operations over software change	0.050	0.037	0.026	0.037	0.010
5	STF5-Software productivity effectiveness due to change of software tools	0.101	0.098	0.113	0.104	0.010
6	STF6-Obsolescence rate of software tools at introduction of new software products	0.180	0.190	0.229	0.199	0.023
7	*STF7-Software reliability of software tools over project change*	0.150	0.166	0.113	0.143	0.023

Table 3 Weightage of lean six sigma software development flexibility variables

Sr. No.	Variables	Software Champion-I weight	Software Champion-II weight	Software Champion-III weight	Average weight	Standard deviation
1	*LSS1-Elimination of waste*	0.337	0.359	0.359	0.351	0.013
2	LSS2-Continuous improvement in software development	0.043	0.046	0.046	0.045	0.002
3	LSS3-Zero defects in software projects	0.090	0.100	0.100	0.096	0.010
4	LSS4-Just in time software deliveries	0.180	0.205	0.204	0.196	0.020
5	LSS5-Pull of raw software data's	0.043	0.046	0.100	0.063	0.032
6	LSS6-Multifunctional software teams	0.178	0.100	0.100	0.126	0.045
7	*LSS7-Decentralization*	0.021	0.022	0.022	0.022	0.001
8	LSS8-Integration functions	0.090	0.100	0.046	0.078	0.028
9	LSS9-Vertical information of software systems	0.021	0.022	0.022	0.022	0.001

A consistency index (CI) and consistency ratio (CR) is also computed to verify numerical and transitive consistency and validity of software experts' judgments for software developers' flexibility, software tools flexibility and lean six sigma software development variables separately. The software data for software developer flexibility, software tool flexibility and lean six sigma software development variables is collected by using a specially designed questionnaire and conducting personnel interaction with software developers' of different levels. Queries have been framed related to these variables to know the response of the company to these variables. Software developer flexibility, software tool flexibility and lean six sigma software development have been measured on 0 to 1 scale. Multiple regression analysis is also carried out to analyze the contribution of two or more resource flexibility variables to the lean six sigma software development. The value of each variable is computed as follows:

$$\text{Value of Xth Variable} = \left(\frac{\sum \text{Sx}}{\text{Smax}}\right) * \text{Wx} \qquad \text{(i)}$$

Where, $\sum Sx = sum\ of\ actual\ grade\ of\ all\ questions\ based\ upon\ Xth\ variable.$

$$\text{Smax} = \text{sum of maximum probable grade of all questions based upon Xth variable.}$$

$$\text{Wx} = \text{weight of Xth variable through FA}HP.$$

$$Software\ Developer\ flexibility\ (SDF) = \sum_{\mathrm{i=0}}^{\mathrm{n}} \text{SDFi} * \text{WFi} \qquad \text{(ii)}$$

Where, SDFi is the ith variable value of software developer flexibility and WFi is its weight computed from FAHP:

$$\text{Software tools flexibility (STF)} = \sum_{\mathrm{i=0}}^{\mathrm{n}} \text{STFi} * \text{WFi} \qquad \text{(iii)}$$

Where, STFi is the ith variable value of software tool flexibility and WFi is its weight computed from FAHP:

$$\text{Lean Six Sigma Software Development (LSS)} = \sum_{I=0}^{N} LSSi * WFi \qquad (iv)$$

Where, LSSi is the ith variable value of lean six sigma software development and WFi is its weight computed from FAHP.

A complete research has been conducted in organizational firm involved in software development of software project module with an objective to analyze minutely the practices being followed for software managing resource (software developer and software tools) flexibility and to look into the difficulties and constraints being faced by the software company while executing lean six sigma software development in their software systems. The status of various variables of information flexibility and lean six sigma software development are computed through questionnaire and personnel interaction.

It is not required that all the variables of software developer flexibility, software tool flexibility and lean six sigma software development have equal contribution towards total software developer flexibility, software tool flexibility and lean six sigma software development, respectively. The impact of one may be greater than other. So, the FAHP has been used for finding the weight of each variable towards software developer flexibility, software tool flexibility and lean six sigma software developments, respectively [35][40]. Describes and elaborates FAHP. Paired comparison is based on the idea that a complex issue can be effectively tested if it is hierarchically decomposed into its elements. The parts are compared with each other. This supplies an opportunity for a pair-wise comparison for developing the structure into NxN reciprocal judgment matrix. In the matrix, one begins with an part on the left and compares how much more significant it is than an part on top. When compared with itself, the ratio is one. When compared with another part, if it is more significant than that part, an integer number, as explained below, is assigned. If, however, it is less significant, then reciprocal of the previous integer number is assigned. In either case reciprocal number is entered in the transpose position matrix. Thus, only N(N-1)/2 judgments are considered where N is the total number. The judgment is to concentrate on only two parts at a time. A scale of 1-9 is used for assigning judgment number according to the following guide lines:

= 1 if i and j are equally significance.

= 3 if i is weakly more significance than j.
=5 if i is strongly more significance than j.
= 7 if i is very strongly more significance than j.
= 9 if i is absolutely more significance than j.

Values 2, 4, 6 and 8 are used to compromise between two judgments.

The weightage of the variables are obtained by computing the eigen vector weights for the judgment matrix. An index of consistency is computed to provide information on how serious is violations of numerical and transitive consistency. The outcomes could be used to seek additional information and re-test the data used in constructing the scale in order to better consistency. Each element in column j of pair wise comparison matrix is divided by the sum of the entries in column j. This gives normalized matrix Xw.

$$\boldsymbol{Xw} = \begin{bmatrix} \frac{\boldsymbol{x11}}{\sum \boldsymbol{xi1}} & \frac{\boldsymbol{x12}}{\sum \boldsymbol{xi2}} & \cdots & \frac{\boldsymbol{x1n}}{\sum \boldsymbol{xin}} \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\boldsymbol{xn1}}{\sum \boldsymbol{xi1}} & \frac{\boldsymbol{xn2}}{\sum \boldsymbol{xi2}} & \cdots & \frac{\boldsymbol{xnn}}{\sum \boldsymbol{xin}} \end{bmatrix}$$

Eigen vector "V" is found out by dividing the sum of all the entries in rows I with "n" no. parameters of normalized matrix:

$$V = \begin{bmatrix} V1 \\ V2 \\ \\ \cdots \\ \cdots \\ \\ Vn \end{bmatrix} = \begin{bmatrix} \frac{\frac{x11}{\sum xi1}}{m} + \frac{\frac{x12}{\sum xi2}}{m} + & \cdots + \frac{\frac{x1n}{\sum xin}}{m} \\ \cdots \quad \cdots & \cdots \quad \cdots \\ \cdots \quad \cdots & \cdots \quad \cdots \\ \frac{\frac{xn1}{\sum xi1}}{m} + \frac{\frac{xn2}{\sum xi2}}{m} + & \cdots + \frac{\frac{xnn}{\sum xin}}{m} \end{bmatrix}$$

It is necessary to check the consistency in the pair wise comparison matrix and the validation of the FAHP. Following procedure is adopted to find out "CI".

Step 1 Calculate "XV":

$$XV = \begin{bmatrix} x11 & x12 & \dots & x1n \\ x21 & x22 & \dots & x2n \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix} X \begin{bmatrix} V1 \\ V2 \\ \\ Vn \end{bmatrix} = \begin{bmatrix} y1 \\ y2 \\ \dots \\ \dots \\ yn \end{bmatrix}$$

The CI is calculated as follows:

$$CI = \frac{\boldsymbol{\lambda max - N}}{\boldsymbol{(N-1)}}$$

Where n is the number of parts being compared and λmax is the largest Eigen value of the judgment matrix:

$$\boldsymbol{\lambda max} = \frac{\frac{1}{N}\sum \boldsymbol{Yi}}{\boldsymbol{Vi}}$$

Step 2 Find the maximum Eigen value, consistency index (CI), consistency ratio (CR), and normalized numbers for each criterion alternative. CR can be computed by dividing CI by the random consistency value for the same size matrix:

$$\boldsymbol{CR} = \frac{CI}{RI}$$

If the maximum Eigen value, CI, and CR are acceptable, then decision is taken based on the normalized numbers; else the procedure is continued till these numbers lies in a useful range CR can be obtained. The satisfactory number of CR is less than ten percent shown by Womack and jones (2005). If CR is not within this range, participants should understand the problem and reconsider their judgment.

The values of CI for software developer flexibility variables is 7.520, 5.060, 6.660 percent and CR is 4.980, 3.350 and 4.410 percent as computed from the response of three software experts, respectively. As shown in Table 1, most important variable of software developer flexibility is "ability of software developers to develop on different software platforms SDF1" with 35.20 percent. This has been followed by "programming skill level of software developers to perform different projects SDF2" with a weightage of 17.10 percent, "software reliability of software developers over project change SDF4" and "ability of software production developers to perform autonomous software maintenance SDF9" with 10.70 percent each. Other variables have got a weightage of less than 10 percent, presenting that they are comparatively less significant in computation of software developer flexibility. Ability of software developers to develop on different software platforms is significant as it enables deployment of developers on platforms and projects other than the ones they normally develop on. In many industries it is a practice now to train the developers on more than one project to achieve this.

The value of consistency index for software tools flexibility variables is 8.920, 5.290, 7.430 percent and consistency ratio is 9.840, 4.010 and 5.620 percent as computed from the response of three software experts, respectively. Table 2 shows the weightage of software tool flexibility variables. The most important variable of software tool flexibility is "ability of software tools to perform diverse set of operations" with 40.400 percent weightage. This has been followed in order by "obsolescence rate of software tools at introduction of new software products -STF6" with 20 percent, next variable is "Software reliability of software tools over software project change - STF7" with 14.2 percent and "software productivity effectiveness due to change of software tool- STF5" with 10.3 percent. Other variables which have got a weightage of less than 10 percent are: "duration and effort required changing the software tools and operations" 7.8 percent, "STF2-software tool setup or changeover" 3.7 percent and "software price effectiveness of operations over software change- STF4" 3.6 percent.

The values of CI for lean six sigma software development variables is 7.510, 7.250, 7.110 percent and CR is 5.170, 5.000 and 4.900 percent as computed from the response of three software experts, respectively. Weightage of lean six sigma software development parameters are shown in Table 3. The most important variable of lean six sigma software development is "elimination of waste-LSS1" with 35.2 percent and followed by "just in time software deliveries- LSS4" with 19.6 percent and "multifunctional software teams-LSS6" with 12.6 percent. Other variables have got a weightage of less than 10 percent, representing that they are comparatively less important in computation of lean six sigma software developments. Elimination of waste is significant as it enables utilization of software resources and maximization of the software productivity of a company. As the values of CI and CR fall within the acceptable limits for all the three software experts and for all software developer flexibility, software tool flexibility and lean six sigma software development variables validates the software reliability and consistency of FAHP. All variables of software tool flexibility, software developer flexibility and lean six sigma software development show significant standard deviation except SDF2. It is clearly illustrated that opinion of all the experts regarding the contribution weightage toward overall software developer flexibility, software tool flexibility and lean six sigma software development by their respective variables is significantly consistent except in case of "skill level of software developers to perform different projects" towards software developer flexibility.

3. RESULTS

A research has been conducted at Software organization. It develops different software projects. The industry is equipped to achieve highly intricate profiles on software developments. The company has also procured advanced technology.

Software organization has a dedicated workforce of 651, out of which seven are senior executives, 47 software engineers, 359 skilled software developers, 99 semi skilled developers, 100 maintenance engineers and 39 are unskilled employees. Table 4 shows the values of various variables of software developer flexibility computed through case study.

Table 4 Evaluation of software developer flexibility at Software Company

Sr. No.	Variable	Software questionnaire	Interaction	Mean	Success in percentage
1	*SDF1*	0.2640	0.2700	0.2670	76.00
2	SDF2	0.1100	0.1230	0.1165	71.00
3	SDF3	0.0230	0.0220	0.0225	77.50
4	SDF4	0.0640	0.0860	0.0750	70.00
5	SDF5	0.0100	0.0300	0.0200	43.00
6	SDF6	0.0700	0.0500	0.0600	85.10
7	*SDF7*	0.0300	0.0300	0.0315	77.50
8	SDF8	0.0530	0.0530	0.0530	80.00
9	SDF9	0.0640	0.0740	0.0690	65.00
10	SDF10	0.0200	0.0200	0.0200	81.00
Software developer flexibility		0.7100	0.7500	0.7300	73.00

76.00 percent of the developers have been trained to develop on different software tools within the company. So in case of absenteeism, there is generally no problem of shifting the developers from one tool to another within the company. 71.00 percent of skilled developers are able to handle different types of projects with same ease and efficiency. Over 70.00 percent of the software developers have multiple skills. Most of the developers are price effective and show good levels of productivity when performing projects other than their routine projects. They have been adequately trained and have positive attitude towards project change. Software training programs are regularly conducted. The aspects of software production developers performing inspection projects and taking up software maintenance are also at satisfactory level. The overall software developer flexibility is shown as 73.10 percent.

Table 5 shows the values of software tools flexibility variables computed through case study. Overall 69.00 percent software tools are general purpose tools and capable of perform different set operations. The special purpose tools also have some degree of software tool flexibility and can handle the parts of various sizes. About 62.00 percent of the software tools have easy setup and changeover from one operation to other. 85.30 percent of tools require low effort and duration to change the tools. 65.10 percent of the same type tools are not price effective due to change of operations. Only 47.70 percent tools are equally software productive and 78.80 percent tools are also capable of producing new software products. 88.30 percent tools are equally reliable over project change. The overall tool flexibility is shown as 72.30 percent.

Table 5 Evaluation of software tool flexibility at Software Company

Sr. No.	Variable	Software questionnaire	Interaction	Mean	Success in percentage
1	*STF1*	0.2830	0.2730	0.2780	69.00
2	STF2	0.0220	0.0240	0.0230	62.00
3		0.0630	0.0700	0.0670	85.30
4	STF4	0.0220	0.0250	0.0230	65.10
5	STF5	0.0410	0.0570	0.0490	47.70
6	STF6	0.1600	0.1550	0.1570	78.80
7	***STF7***	**0.1300**	**0.1240**	**0.1250**	**88.30**
Software tools flexibility		**0.7160**	**0.7300**	**0.7230**	**72.30**

Table 6 Evaluation of lean six sigma software development at Software Company

Sr. No.	Variable	Software questionnaire	Interaction	Mean	Success in percentage
1	***LSS1***	**0.2990**	**0.2750**	**0.2870**	**81.70**
2	**LSS2**	**0.0360**	**0.0380**	**0.0370**	**81.70**
3	**LSS3**	**0.0580**	**0.0660**	**0.0620**	**64.20**
4	**LSS4**	**0.1470**	**0.1530**	**0.1500**	**76.70**
5	**LSS5**	**0.0420**	**0.0430**	**0.0425**	**67.40**
6	**LSS6**	**0.1010**	**0.1060**	**0.1035**	**82.00**
7	***LSS7***	**0.0180**	**0.0160**	**0.0170**	**77.20**
8	**LSS8**	**0.0680**	**0.0550**	**0.0615**	**78.60**
9	**LSS9**	**0.0150**	**0.0160**	**0.0155**	**69.00**
Lean Six Sigma software development		**0.7820**	**0.7700**	**0.7800**	**77.50**

Table 6 shows the values of different variables of lean six sigma software development computed through case study. G-soft achieved 77.50 percent of overall lean six sigma software development through eliminating the 81.70 percent wastes from the software development system. The results also depict that about 80 percent of the total seven wastes are eliminated at Software Company. 81.70 percent continuous improvement is achieved through adopting the concept of software quality circles, kaizen software teams, conducting software training programs. 64.20 percent zero defects is achieved through executing Six Sigma methodologies. 76.70 percent just in time deliveries is achieved through reducing the one day inventory of finished software products, three days inventory of bought out parts and six days inventory of raw data's. 67.40 percent pull of raw data's is achieved through utilizing Kanban cards for data movement and software production purposes. Size of batch is also reduced over the passage of time. 82.00 percent multifunctional software teams are software developed for proper manpower utilization. 77.20 percent decentralization is achieved through delegating the powers at various levels. 78.60 percent functions are integrated through software quality policy. 69.00 percent vertical information systems are adopted in the company.

The analysis of multiple regressions has been identifying set of parameters which conjointly contribute significantly towards the criterion parameter. A stepwise multiple linear regression analysis enables to know the most relevant parameters which account for maximum variance in the criterion from the total set of parameters. Multiple regression analysis is a method of analyzing the collective and separate contribution of two or more independent parameters "X" to the variation of dependent parameters "Y" illustrated by Zhang et al (2012).

Table 7 Results for the analysis of multiple linear regressions

Parameters criterion	Prediction parameters	Selected parameters	Multiple correlation (R)	% of Attribution (R^2)	F value	Significant
LSS1	*All variables of software resource flexibility*	SDF1,SDF3,SDF5,STF4	0.7810	61.00	18.51811	0.00000
LSS2		SDF1,SDF2,SDF10,STF1	0.7220	52.12	12.40400	0.00000
LSS3		SDF1,SDF2	0.6100	37.21	13.75512	0.00000
LSS4		SDF1,SDF2,SDF5,SDF10,STF1	0.8200	67.24	16.7514	0.00000
LSS5		SDF1,SDF4,SDF5,SDF10,STF1,STF2	0.8410	70.72	17.72300	0.00000
LSS6		SDF1,SDF2,SDF10,STF1,STF5	0.8520	72.59	24.14000	0.00000
LSS7		SDF1,SDF8,STF1,STF7	0.7620	58.06	16.39700	0.00000
LSS8		SDF1,SDF2,SDF10,STF3,STF6	0.8230	67.73	19.11911	0.00000
LSS9		SDF1,SDF4,SDF5,STF3	0.7310	53.44	13.30000	0.00000
LSS		SDF1,SDF2,SDF5,SDF10,STF1	0.8720	76.04	29.53000	0.00000

Table 7 represents that the results of analysis of multiple linear regression of various variables of software resource flexibility towards lean six sigma software development variables and overall lean six sigma software developments.

4. DISCUSSION

Out of seventeen parameters only four parameters SDF1, SDF3, SDF5 and STF4 have emerged to be significant predictors of LSS1. Conjoint predictive value of these resource flexibility variables for elimination of waste is 61.00 percent. The analysis depicts that the conjoint predictive value of SDF1, SDF2, SDF10 and STF1 for LSS2 is 52.12 percent. So 52.12 percent of whatever leads to increase in continuous improvement is attributable to these four variables. Two variables SDF1, SDF2 contribute significantly towards achieving defects free software. The value of R^2 illustrates that conjoint predictive value of these two variables towards zero defects is 37.21 percent. Five variables SDF1, SDF2, SDF5, SDF10 and STF1 significantly contribute 67.24 percent towards just in time deliveries LSS4. Six parameters SDF1, SDF4, SDF5,SDF10, STF1 and STF2 have emerged to be significant predictors of pull of raw data LSS5. Conjoint predictive value of these variables for pull of raw data's is 70.72 percent. Five variables SDF1, SDF2, SDF10, STF1and STF5 have emerged to be significant predictors of multifunctional software teams LSS6. Collective conjoint predictive value of these variables for multifunctional software teams is 72.59 percent. Four variables, SDF1, SDF8, STF1, STF7 have emerged to be significant predictors of decentralization LSS7. Collective conjoint predictive value of these variables for decentralization is 58.06 percent. Five variables SDF1, SDF2, SDF10, STF1 and STF5 have emerged to be significant predictors of LSS8. Collective conjoint predictive value of these variables for LSS8 is 67.73 percent. Four variables SDF1, SDF4, SDF5 and STF3 have emerged to be significant predictors of LSS9. Collective conjoint predictive value of these variables for vertical information systems is 53.44 percent. Five variables SDF1,SDF2,SDF5,SDF10 and STF1 have emerged to be significant predictors of overall lean six sigma software development LSS. Collective conjoint predictive value of these variables for lean six sigma software development is 76.04 percent. It is inferred that 76.04 percent of whatever leads to support lean six sigma software development can be endorsed to SDF1, SDF2, SDF5, SDF10 and STF1 significantly.

5. CONCLUSION

The present research provides the most important parameters to the software managers for evaluate the status of software resource flexibility and lean six sigma software development. The overall software developer flexibility achieved by the company is 73.00 percent. The software company has focused on the software productivity effectiveness due to change as highest level as of 85.10 percent through centering on the regular software training of developers at a level of 81.00 percent about the new software, methods and to handle multitask jobs at a level of 80.00 percent. Price effectiveness of software developers over project change is attained at a level of 77.50 percent. The developers cooperate 77.50 percent in achieving software production targets and 76.00 percent developers are able to develop on different platforms. Programming skill level of developers to perform different project is 71.00 percent while 70.00 percent developers are reliable over project change. 65.00 percent of software production developers are able to do autonomous software maintenance however only 43.00 percent developers have positive attitude towards change. The overall software tool flexibility achieved by the company is72.30percent. 88.30 percent tools are reliable over project change and on 85.30 percent tools and operations can be easily changed without wasting much duration and effort. 78.80 percent tools do not obsolete with introduction of new software products and 69.00 percent tools are also capable to perform diverse set of operations. 65.10 percent tools are not price effective for same type of operations and 62.00 percent tools can be set up or changeover from one operation to other with same ease. Software tool productivity varies from tool to tool and only 47.70 percent tools are capable to perform at uniform software productivity. The software company has realized 77.50 percent lean six sigma software developments. The software company successfully develops 82.00 percent multifunctional software development teams and attains a level of 81.70 percent for continuous improvement to achieve lean six sigma software development. The company also eliminates 81.70 percent waste from different areas through integrating 78.60 percent functions, 77.20 percent decentralization, 69.00 percent vertical information systems. The company has also executed 76.70 percent just in time through attaining 67.40 percent pull of raw data's. The firm has achieved 64.20 percent zero defects concept. Conjoint contribution of different variables of software resource flexibility towards different variables of lean six sigma software development is positive. It is inferred that 76.04 percent of lean six sigma software development is endorsed by software resource flexibility. Thus, the analysis of multiple regression validate the positive and significant impact of software resource flexibility variables on lean six sigma software development.

The present research highlights the fact that it is possible to plan for software resource flexibility keeping lean six sigma software development in mind. Therefore, it is concluded that there is a broad scope to focus upon the strategies which can change the attitude of software developers towards change positively. More software tools should be installed which are having uniform software productivity even change of operations and operators. There is also a tremendous scope to achieve higher degree of lean six sigma software development by executing zero defects concept holistically. Present research will help the software practitioners to identify the meager points in software development system to manage resource flexibility by executing the different variables hierarchically according to their importance and realization of price cutting through optimum utilization of resources. The present research is limited to only software developer and software tools as the elements of software resource flexibility. Other resources may also be included to calculate overall software resource flexibility in the future research works.

REFERENCES

1. Abbas, A. (1996), "Organizational development in the Arab world", The Journal of Management Development, Vol. 15 No. 5, pp. 4-12.
2. Achanga, P., Shehab, E., Roy, R. and Nelder, G. (2006), "Critical success factors for lean implementation within SMEs", Journal of Manufacturing Technology Management, Vol. 17 No. 4, pp. 460-471.
3. Angelis, J., Conti, R., Cooper, C. and Gill, C. (2011), "Building a high-commitment lean culture", Journal of Manufacturing Technology Management, Vol. 22 No. 5, pp. 569-586.
4. Antony, J. (2006), "Implementing the lean Sigma framework in an Indian SME: a case study", Production Planning and Control, Vol. 17 No. 4, pp. 407-423.
5. Asmild, M., J.C. Paradi, and D.N. Reese, (2006). Theoretical Perspectives of Trade-off Analysis Using DEA, Omega, 34(4): pp. 337-343.
6. Atkinson, P. (2010), "Lean is a cultural issue", Management Services, Vol. 54, pp. 35-44.
7. Buglione, L., Trudel, S. (2010), "Guideline for sizing Agile projects with COSMIC", In: Proceedings of the IWSM / MetriKon / Mensura 2010, Stuttgart, Germany
8. Bhasin, S. (2011), "Performance of organizations treating lean as an ideology", Business Process Management Journal, Vol. 17 No. 6, pp. 986-1011.
9. Bergmiller, G.G. and McCright, P.R. (2009), "Parallel models for lean and green operations", paper presented at Industrial Engineering Research Conference, Miami, FL.

10. Bonavia, T. and Marin, J.A. (2006), "An empirical study of lean production in the ceramic tile industry in Spain", International Journal of Operations & Production Management, Vol.26 No. 5, pp. 505-531.
11. Boynton, A.C. and Zmud, R.W. (1984), "An assessment of critical success factors", Sloan Management Review, Vol. 25 No. 4, pp. 17-29.
12. Chen, C., (2000). Extensions of the TOPSIS for Group Decision-Making under Fuzzy Environment. Fuzzy Sets and Systems, 114(1), pp. 1-9.
13. Chin, K.S. and Pun, K.F. (2002), "A proposed framework for implementing TQM in Chinese organizations", International Journal of Quality and Reliability Management, Vol. 19 No. 3, pp. 272-294.
14. Chong, V.K. and Rundus, M.J. (2004), "Total quality management, market competition and organizational performance", The British Accounting Review, Vol. 36, pp. 55-172.
15. Conner, G. (2001), Lean Manufacturing for the Small Shop, Society of Manufacturing Engineers, Dearbon, MI.
16. Coronado, R.B. and Antony, J. (2002), "Critical success factors for the successful implementation of Six Sigma projects in organizations", The TQM Magazine, Vol. 14 No. 2, pp. 92-99.
17. Creveling, C.M., Slutsky, J.L., Antis, D. (2003), Design for Six Sigma, Prentice Hall, Princeton, NJ
18. Dahlgaard, J.J. and Dahlgaard-Park, S.M. (2006), "Lean production, Six Sigma quality, TQM and company culture", The TQM Magazine, Vol. 18 No. 3, pp. 263-281.
19. Eltony, M.N. (2007), "The economic development experience of Kuwait: some useful lessons", Journal of Economic and Administrative Sciences, Vol. 23 No. 1, pp. 77-102.
20. Furlan, A., Vinelli, A. and Dal Pont, G. (2011), "Complementory and lean manufacturing bundles: an empirical analysis", International Journal of Operations & Production Management, Vol. 31 No. 8, pp. 835-850.
21. Fehlmann, Th. (2004), "Six Sigma for Software", in: Proceedings of the 1st SMEF Conference, Rome, Italy
22. George, M. ed. (2010), The Lean Six Sigma Guide to Doing More with Less, John Wiley & Sons, Hoboken, NJ
23. Kundu, G. and Manohar, B.M. (2012), "Critical success factors for implementing lean practices in it support services", International Journal for Quality research, Vol. 6 No. 4, pp. 301-312.
24. Kunnanatt, J.T. (2007), "Impact of ISO 9000 on organizational climate", International Journal of Manpower, Vol. 28 No. 2, pp. 175-192.
25. Kwak Y H, & Anbari FT. (2006). Benefits, obstacles and future of Six Sigma approach. Technovation, 26(5-6): 708-715.
26. Liker, J.K. and Hoseus, M. (2010), "Human resource development in Toyota culture", International Journal of Human Resources Development and Management, Vol. 10 No. 1, pp. 34-50.
27. Maroofi, F. and Dehghan, S. (2012), "Performing lean manufacturing system in small and medium enterprises", International Journal of Academic Research in Accounting, Finance and Management Sciences, Vol. 2 No. 3, pp. 156-163.
28. Nordin, N., Deros, B.M. and Abd Wahab, D. (2012), "A framework for managing change in lean manufacturing implementation", International Journal of Services and Operations Management, Vol. 12 No. 1, pp. 101-117.
29. Panizzolo, R. (1998), "Applying the lessons learned from 27 lean manufacturers: the relevance of relationships management", International Journal of Production Economics, Vol. 55 No. 3, pp. 223-240.
30. Poppendieck, M. & T. (2007), Implementing Lean Software Development: From Concept to Cash. Addison-Wesley Professional, Boston, MA
31. Pool, A., Wijngaard, J. and Zee, D.J. (2011), "Lean planning in the semi-process industry, a case study", International Journal of Production Economics, Vol. 131, pp. 194-203.
32. Roriz, H. (2009), "Can Scrum Support Six Sigma?", in: Scrum Alliance, http://www.scrumalliance.org/articles/161-can-scrum-support-six-sigma [seen 26-feb-11].
33. Ross, T.J., (1995). Fuzzy Logic with Engineering Applications, McGraw-Hill, Inc.
34. S.M. Kazemi, (16-18 July 2012). Six Sigma project selection by using a fuzzy multiple criteria decision making approach a case study in poly acryl corp. CIE42 proceedings, Cape Town, South Africa 2012 CIE & SAIIE.
35. Staats, B.R., Brunner, D.J. and Upton, D.M. (2011), "Lean principles, learning and knowledge work: evidence form a software services provider", Journal of Operations Management, Vol. 29, pp. 376-390.
36. Snee, Ronald: Leading Six Sigma: A Step-by-Step Guide Based on Experience with GE and Other Six Sigma Companies, Financial Times Prentice Hall 2002
37. Stone, K.B. (2012), "Four decades of lean: a systematic literature review", International Journal of Lean Six Sigma, Vol. 3 No. 2, pp. 112-132.
38. Upadhye, N., Deshmukh, S.G. and Garg, S. (2010a), "Lean manufacturing system for medium size manufacturing enterprises an Indian case", International Journal of Management Science and Engineering Management, Vol. 5 No. 5, pp. 362-375.
39. Upadhye, N., Deshmukh, S.G. and Suresh, G. (2010b), "Lean manufacturing for sustainable development", Global

Business and Management Research: An International Journal, Vol. 2 No. 1, pp. 125-137.

40. Vanichchinchai, A. (2012), "The relationship between employee involvement, partnership management and supply performance: findings from a developing country", International Journal of Productivity and Performance Management, Vol. 61 No. 2, pp. 157-172.
41. Vinodh, S. and Balaji, S. (2011), "Fuzzy logic based Leanness assessment and its decision support system", International Journal of Production Research, Vol. 49, pp. 40-67.
42. Wilson, L. (2009), How to Implement Lean Manufacturing, McGraw-Hill, New York, NY
43. White, R.E., Pearson, J.N. and Wilson, J.R. (1999), "JIT manufacturing: a survey of implementation in small and large US manufacturers", Management Science, Vol. 45 No. 1, pp. 1-15.
44. Womack, J. and Jones, D. (2003), Lean Thinking: Banish Waste and Create Wealth in Your Corporation, 2nd ed., The Free Press, New York, NY.
45. Womack, J. and Jones, D. (2005), Lean Solutions: How Companies and Customers Can Create Value and Wealth Together, Lean Enterprise Institute, Brookline, MA.
46. Wong, M. (2007), "The role of culture in implementing lean production system", in Olhager, J. and Persson, F. (Eds), IFIP International Federation for Information Processing, Boston, MA, Vol. 246, pp. 413-422.
47. Yang, M.G., Hong, P. and Modi, S.B. (2011), "Impact of lean manufacturing and environmental management on business performance: an empirical study of manufacturing firms", International Journal of Production Economics, Vol. 129, pp. 251-261.
48. Zhou, B. (2012), "Lean principles, practices, and impacts: a study on small and medium-sized enterprises", Annals of Operations Research, July, pp. 1-18.
49. Zhang, Q., bbas, J., Zhu, X. and Shah, M. (2012), "Critical success factors for successful Lean Six Sigma implementation in Pakistan", Inter disciplinary Journal of Contemporary Research in Business, Vol. 4 No. 1, pp. 117-124