Gap Analysis between Theory and Practice of Software Testing Methodologies in Indian IT Industry

Dilbag Singh, Ph.D
Dept. of CSA
Ch. Devi Lal University, Sirsa

Pradeep Kumar, Ph.D
Dept. of IT
LBS Inst. of Mgmt., Delhi

Sanjeev Kumar
Dept. of CSA
Ch. Devi Lal University, Sirsa

ABSTRACT
There are broadly two approaches for software testing-manually or automatically. Manual testing uses the knowledge of the tester to target testing of the system that is assumed to be more error-prone. Automated testing can perform a large number of tests in little time. Software testing is taught to the computer/IT graduates in Indian universities. Tools for manual and automatic testing are available in the market to enhance the productivity and reliability of the testing process.

The survey focused on three major aspects of software testing, namely software testing education/ training, testing methodologies/ techniques, and automated testing tools. Based on the survey results, current practices in software testing are reported, as well as some observations and recommendations for the future of software testing in India for academia and industry.

Keywords: Software testing, Manual testing, Automated testing, IT Industry

1. INTRODUCTION
Software testing is the process of executing a program with the intention of finding errors in the code [4]. It is the process of exercising or evaluating a system or system component by manual or automatic means to verify that it satisfies specified requirements or to identify differences between expected and actual results [8]. Software Testing should not be a distinct phase in system development but should be applicable throughout the design development and maintenance phases. Software testing is the process of executing software in a controlled manner, in order to answer the question. Does the software behave as specified? One way to ensure system's responsibility is to extensively test the system [9]. Since software is a system component it requires a testing process also. The overall testing process benefits from the strengths of both manual and automated testing [10].

With a manual strategy, testers prepare a test suite that generates test cases from the program's specification (black box) or its actual text (white box) [7]. An automated testing strategy tries to remove the tediousness of the process by relying on a software tool. Automated and manual strategies are often thought of as completely distinct, and usually supported by different tools. Manual tests are good for capturing deep or special cases, which automated tests might not guess [11]. But they cannot yield extensive coverage. Automated tests are good at breadth but much less at depth. Manual unit testing has established itself as an integral part in modern software development [12]. Automated testing automates not only test case execution, but also test case generation and test result verification. A fully automated testing system is able to test software without any user intervention [1]. Automated testing requires less effort on the developer's side, but it cannot fully replace manual unit testing. Developers are better at setting up complex input data and at finding interesting test cases [2].

There were a number of reasons for conducting this survey: Firstly, to determine whether existing training courses in software testing taught in the workplace or in similar study at Indian universities adequately cover the types of testing methodologies and skills that industry requires. Secondly, to identify gap between theory and practices for both manual and automated testing in IT industry. Finally, the survey may provide indications of future research directions. The observations reported in this paper are based on 107 respondents for manual testing and 104 respondents for automation testing who have completed the questionnaire successfully. The respondents were from various IT companies from NCR region. Despite the relatively small sample population in the survey, the consistency of the data obtained heightened our confidence to report the observations in this paper.

2. METHOD DESCRIPTION
2.1 Survey Description
The survey targeted senior employees involved with testing in software development organizations. Requests were addressed to software testing or quality mangers of the IT companies in NCR region to understand their testing environments and experiences.

Three major areas of software testing related activities like software testing education/ training, software testing methodologies/techniques, automated software testing tools etc. were investigated by the survey. The information sought can be summarized as follows [5].

Software Testing Education/ Training: It determined the extent to which they have learnt the knowledge of softer testing through academics and training by the organizations for their employees. The usages of various sources of training courses were also queried [3].

Software Testing Methodologies/Techniques: The extent to which software testing methodologies and general testing techniques are used in the industry and the current practices of those organizations adopting structured methodologies and techniques in software testing were investigated [1].

Automated Software Testing Tools: Questions relating to the extent to which automated testing tools are used in industry, including commercial and in-house developed tools, were revealed. The level of satisfaction with such tools was assessed by querying the respondents' belief that the quality of developed software was being improved by the use of such tools [6].

2.2 Survey Method

A questionnaire comprised 25 questions in each of manual and automated testing was used. Survey interviews were conducted face to face, over the telephone, via email attachment. To allow for more flexible arrangements, some respondents were invited to complete the online questionnaire at our survey website. In all cases, printed or verbal explanatory notes were provided to respondents to ensure consistent interpretation of the terminologies and questions in the questionnaire. Confidentiality and privacy were assured to all individuals returning the questionnaire and the organization that they represented.

2.3 Sample Selection and Responses

2.3.1 Sample Selection

Our survey targeted the population at the organizational level. A draft questionnaire of the survey was trialed against a small group of organizations, and a number of adjustments were made based on the experiences and feedback we gathered from respondents. As a result, we aimed at targeting four different types of participants in this survey. The first preference was test managers, the second was a member of the test team, thirdly a software quality manager, and finally a general IT professional having testing experience. This allowed us to deal with situations where there was no specific individual responsible for testing in the organization.

The both questionnaire consists of five-level Likert-scale response alternatives: Strongly disagree, Disagree, Neutral, Agree, and Strongly Agree. As a result, 107 respondents for manual testing and 104 respondents for automation testing completed the questionnaire successfully in the survey. This is a relatively low response rate, given the large number of organizations that were invited to participate in the survey, and the large estimated size of the population. The questionnaires were pre-tested. The purpose was to test the instruments for validity & reliability to determine how realistic the questions were to the ability of users. Minor changes were made after the pre-test, based on feedback we gathered from respondents. Cronbach's Alpha test was used on the data. It provides a measure of the internal consistency of a test or scale. It is expressed as a number between 0 and 1. Value of alpha for manual testing was .683 and for automated testing it was .717 calculated using SPSS software.

2.3.2 Responses

The responses of survey on manual testing are provided in table-1 and responses of survey on automated testing are provided in table-2.

3. ANALYSIS OF RESULT

3.1 Gap Analysis of Theory and Practice for Manual Testing

Data of 107 respondents was analyzed by applying factor analysis using SPSS software. The purpose of factor analysis is to reduce multiple variables to a lesser number of underlying factors that are being measured by the variables. To test the appropriateness of factor analysis technique Kaiser-Meyer-Olkin (KMO) measure of sample adequacy is used. KMO compares the magnitude of observed correlation coefficients to magnitude of partial correlation coefficients. KMO identified values greater than 0.5 might desirable. In our analysis, the value of KMO comes out is .727 with significance .000, which is acceptable.

Principal Component Analysis is a dimension reduction technique to analyze factors responsible for gap in manual software testing. A set of factors is identified based upon possible information available in the data. Nine factors are extracted on the basis of Principal Component Analysis. These nine factors describe the relationship among variables in a best way showing cumulative %age of variances, it can be observed that nine factors identifying 71.192 % of variance contributed by first component is 19.482 followed by second (27.486), third (34.793), fourth (41.091), fifth (47.314), sixth (53.467), seventh (59.598), eighth (65.601) and last ninth is (71.192) [Table-3]. In rotated component matrix, a value greater than .500 is considered for analysis across these nine factors against 25 question statements

3.1.1 Manual Testing Factors [Table-4]:

Factor 1: In factor 1, four variables found to be high factor loading i.e. .801(Final project training is actually done at s/w industry in testing domain), .505 (Manual s/w testing concepts used in industry are same as you studied), .832(For manual testing job, training is not required from industry /private institute), and .847(In house training is not required for manual testing). So these four variables can be clubbed and factor may be named as "There is no gap between theoretical and actual implementation of manual s/w testing".

Factor 2: In factor 2, two variables found to be high factor loading i.e. .889(Implementation (Practical) of software testing strategies was part of curriculum), and .666(Marks of implementation (practical's) of ST added to marks of final degree). So these two variables can be clubbed and factor may be named as "Software testing syllabus includes practical of manual testing techniques and marks are added to degree".

Factor 3: In factor 3, two variables found to be high factor loading i.e. .847(Implementation (Practical) of software engineering concepts was part of curriculum), and .839 (Marks of implementation (practical) of SE added to marks of final degree). So these two variables can be clubbed and factor may be named as "Software engineering practical are not adequate in curriculum but marks are added to degree".

Factor 4: In factor 4, two variables found to be high factor loading i.e. .736(Academic curriculum covers in depth knowledge of Software engineering (SE)), and .708(SE curriculum covers architectural design of software). So these two variables can be clubbed and factor may be named as "Software engineering has theoretical foundation in curriculum".

Factor 5: In factor 5, two variables found to be high factor loading i.e. .686(SE syllabus covers basic knowledge of software testing), and .660(ST was a separate subject with focus on manual testing in academic curriculum). So these two variables can be clubbed and factor may be named as "Software testing has theoretical foundation in curriculum"

Table-1: Responses of survey on manual testing

	Questions	Strongly Disagree		Disagree		Neutral		Agree		Strongly Agree	
		Frequency	%	Frequency	%	Frequency	%	Frequency	%	Frequency	%
1	Academic curriculum provides only basic knowledge of Software testing (ST).	10	9.3	61	57.0	15	14.0	15	14.0	6	5.6
2	Academic curriculum covers in depth knowledge of Software engineering (SE).	5	4.7	6	5.6	22	20.6	58	54.2	16	15.0
3	Implementation (Practicals) of software engineering concepts was part of curriculum.	11	10.3	32	29.9	22	20.6	35	32.7	7	6.5
4	Marks of implementation (practical) of SE added to marks of final degree.	8	7.5	29	27.1	28	26.2	35	32.7	7	6.5
5	SE syllabus covers basic knowledge of software testing.	5	4.7	5	4.7	5	4.7	47	43.9	45	42.1
6	Academic curriculum of SE meets the requirements of present s/w industry	3	2.8	4	3.7	14	13.1	43	40.2	43	40.2
7	ST was a separate subject with focus on manual testing in academic curriculum.	2	1.9	1	0.9	14	13.1	47	43.9	43	40.2
8	ST syllabus provided knowledge of black box testing.	8	7.5	19	17.8	21	19.6	38	35.5	21	19.6
9	ST syllabus provides knowledge of white box testing.	6	5.6	22	20.6	23	21.5	37	34.6	19	17.8
10	In depth knowledge of complete software testing was part of curriculum.	4	3.7	1	0.9	21	19.6	45	42.1	36	33.6
11	Final project training is actually done at s/w industry in testing domain	14	13.1	26	24.3	22	20.6	25	23.4	20	18.7
12	Project classes and practical viva was under control of industry expert.	5	4.7	13	12.1	13	12.1	44	41.1	32	29.9
13	Implementation (Practical) of software testing strategies was part of curriculum.	6	5.6	25	23.4	27	25.2	36	33.6	13	12.1
14	SE curriculum covers architectural design of software.	4	3.7	5	4.7	6	5.6	52	48.6	40	37.4
15	Regression testing is part of theory/software maintenance.	5	4.7	5	4.7	5	4.7	50	46.7	42	39.3
16	Manual s/w testing concepts used in industry are same as you studied.	10	9.3	14	13.1	16	15.0	41	38.3	26	24.3
17	More emphasis on programming languages than SE in curriculum	5	4.7	17	15.9	21	19.6	43	40.2	21	19.6
18	Live projects of ST were included in curriculum through industry collaboration.	22	20.6	54	50.5	20	18.7	10	9.3	1	0.9
19	Implementation (Practical) of software testing strategies should be part of curriculum.	12	11.2	29	27.1	7	6.5	29	27.1	30	28.0
20	Marks of implementation (practical's) of ST added to marks of final degree.	10	9.3	14	13.1	26	24.3	35	32.7	22	20.6
21	Able to join software testing job directly after completion of academics.	14	13.1	33	30.8	32	29.9	21	19.6	7	6.5
22	ST syllabus provides knowledge of object oriented testing.	3	2.8	8	7.5	19	17.8	45	42.1	32	29.9
23	SE concepts used in industry are same as you studied.	4	3.7	3	2.8	21	19.6	41	38.3	38	35.5
24	For manual testing job, training is required from industry /private institute.	24	22.4	40	37.4	5	4.7	21	19.6	17	15.9
25	In house training is required for manual testing.	30	28.0	34	31.8	7	6.5	20	18.7	16	15.0

Factor 6: In factor 6, one variable found to be high factor loading i.e. .653(Live projects of ST were not included in curriculum through industry collaboration). So this variable can be named as "Live projects of ST are not included in curriculum through industry collaboration".

Factor 7: In factor 7, two variables found to be high factor loading i.e. .708(ST syllabus provided knowledge of black box testing), and .528(ST syllabus provides knowledge of white box testing). So these two variables can be clubbed and factor may be named as "Software testing syllabus includes theory of manual testing techniques".

Table-2: Responses of survey on automated testing

	Questions	Strongly Disagree		Disagree		Neutral		Agree		Strongly Agree	
		Frequency	%	Frequency	%	Frequency	%	Frequency	%	Frequency	%
1	Academic curriculum meets the requirements of present s/w industry.	14	13.5	43	41.3	30	28.8	12	11.5	5	4.8
2	Academic curriculum provides basic knowledge of software testing (ST).	7	6.7	21	20.2	28	26.9	36	34.6	12	11.5
3	In depth knowledge of software testing was part of curriculum.	4	3.8	25	24.0	19	18.3	42	40.4	14	13.5
4	ST should be included as a separate subject in academic curriculum.	2	1.9	13	12.5	35	33.7	39	37.5	15	14.4
5	Automation testing was separate subject in curriculum.	18	17.3	46	44.2	23	22.1	15	14.4	2	1.9
6	Automation testing should be part of software testing.	4	3.8	9	8.7	8	7.7	42	40.4	41	39.4
7	Implementation (Practical) of software testing strategies was part of curriculum.	7	6.7	17	16.3	21	20.2	36	34.6	23	22.1
8	Marks of implementation (practical's) of ST added to marks of final degree.	2	1.9	23	22.1	28	26.9	33	31.7	18	17.3
9	Demo version of automated testing tools were used for practical of software testing.	14	13.5	27	26.0	29	27.9	25	24.0	9	8.7
10	Automation testing was introductory part of curriculum.	26	25.0	47	45.2	20	19.2	9	8.7	2	1.9
11	Automation testing was part of software testing.	26	25.0	53	51.0	9	8.7	11	10.6	5	4.8
12	ST syllabus provides knowledge of various testing techniques.	5	4.8	9	8.7	23	22.1	34	32.7	33	31.7
13	Automation testing tools learned by private training institute.	12	11.5	16	15.4	10	9.6	41	39.4	25	24.0
14	Automation testing was part of software engineering subject.	34	32.7	49	47.1	8	7.7	7	6.7	6	5.8
15	In depth knowledge of automated testing tools was part of curriculum.	21	20.2	40	38.5	30	28.8	8	7.7	5	4.8
16	Implementation (Practical) of software testing strategies should be part of curriculum.	21	20.2	53	51.0	19	18.3	7	6.7	4	3.8
17	Project classes and practical viva must under control of industry expert.	27	26.0	42	40.4	13	12.5	15	14.4	7	6.7
18	Live projects of automated ST should be included in curriculum through industry collaboration.	8	7.7	21	20.2	28	26.9	31	29.8	16	15.4
19	Automation testing tools learned by in house training by company.	8	7.7	10	9.6	24	23.1	35	33.7	27	26.0
20	Licensed tools of automated testing with full functionality were part of curriculum.	10	9.6	21	20.2	31	29.8	32	30.8	10	9.6
21	Automation testing tools learned by self-study /google /you tube.	9	8.7	13	12.5	24	23.1	30	28.8	28	26.9
22	Final project training is actually done at s/w industry in testing domain	8	7.7	20	19.2	24	23.1	35	33.7	17	16.3
23	ST syllabus provides knowledge of object oriented testing.	6	5.8	8	7.7	8	7.7	50	48.1	32	30.8
24	Industries provide in house training for self-developed testing tools.	4	3.8	11	10.6	10	9.6	45	43.3	34	32.7
25	All automation testing tools could not be included in curriculum.	3	2.9	6	5.8	14	13.5	44	42.3	37	35.6

Table-3: Total Variance Explained

Component	Initial Eigenvalues			Extraction Sums of Squared Loadings			Rotation Sums of Squared Loadings		
	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %
1	6.185	24.741	24.741	6.185	24.741	24.741	4.871	19.482	19.482
2	2.179	8.714	33.455	2.179	8.714	33.455	2.001	8.004	27.486
3	1.748	6.993	40.448	1.748	6.993	40.448	1.827	7.307	34.793
4	1.633	6.534	46.981	1.633	6.534	46.981	1.574	6.297	41.091
5	1.458	5.834	52.815	1.458	5.834	52.815	1.556	6.223	47.314
6	1.312	5.247	58.062	1.312	5.247	58.062	1.538	6.153	53.467
7	1.207	4.828	62.890	1.207	4.828	62.890	1.533	6.131	59.598
8	1.055	4.219	67.109	1.055	4.219	67.109	1.501	6.003	65.601
9	1.021	4.084	71.192	1.021	4.084	71.192	1.398	5.592	71.192
1	.868	3.470	74.663						
11	.836	3.344	78.007						
12	.768	3.072	81.079						
13	.670	2.682	83.761						
14	.614	2.455	86.216						
15	.585	2.341	88.557						
16	.507	2.029	90.587						
17	.482	1.930	92.516						
18	.379	1.515	94.032						
19	.322	1.288	95.320						
20	.265	1.059	96.379						
21	.247	.988	97.367						
22	.201	.805	98.172						
23	.184	.735	98.907						
24	.141	.563	99.470						
25	.132	.530	100.000						

Extraction Method: Principal Component Analysis.

Factor 8: In factor 8, two variables found to be high factor loading i.e. .840(Academic curriculum of SE meets the requirements of present s/w industry), and .573(SE concepts used in industry are same as you studied). So these two variables can be clubbed and factor may be named as "Academic curriculum of SE meets the requirements of present s/w industry and SE concepts used in industry are same as studied".

Factor 9: In factor 9, one variable found to be high factor loading i.e. .663(More emphasis on programming languages than SE in curriculum). So this variable can be named as "More emphasizes on programming language".

3.2 Gap Analysis of Theory and Practice for Automated Testing

Data of 104 respondents was analyzed by applying factor analysis using SPSS software. The purpose of factor analysis is to reduce multiple variables to a lesser number of underlying factors that are being measured by the variables. To test the appropriateness of factor analysis technique Kaiser-Meyer-Olkin (KMO) measure of sample adequacy is used. KMO compares the magnitude of observed correlation coefficients to magnitude of partial correlation coefficients. KMO identified values greater than 0.5 might desirable. In our analysis, the value of KMO comes out is .739 with significance .000, which is acceptable.

Principal Component Analysis is a dimension reduction technique to analyze factors responsible for gap in manual software testing. A set of factors is identified based upon possible information available in the data. Seven factors are extracted on the basis of Principal Component Analysis. These seven factors describe the relationship among variables in a best way showing cumulative %age of variances, it is observed that seven factors identifying 68.146 % of variance contributed by first component is 18.391 followed by second is 30.348, third is 39.247, fourth is 47.957, fifth is 55.767, sixth is 62.924, seventh is 68.146 [Table-5]. In rotated component matrix, a value greater than .500 is considered for analysis across these seven factors against 25 question statements.

3.1.2 Automated Testing Factors [Table-6]

Factor 1: In factor 1, five variables found to be high factor loading i.e. .592(Automation testing should be part of software testing), .688(Automation testing tools learned by private training institute), .690(Automation testing tools learned by in house training by company), .694(Automation testing tools learned by self-study /google /you tube), and .806(Industries provide in house training for self-developed testing tools). So these five variables can be clubbed and factor may be named as "There is gap between theoretical and actual implementation of automation s/w testing and automated testing tools are learned during job not in academic".

Table-4: Rotated Component Matrix(Rotation converged in 16 iterations)

	Component								
	1	2	3	4	5	6	7	8	9
Academic curriculum provides only basic knowledge of Software testing (ST).	-.054	.071	.159	-.222	-.325	.350	.421	-.204	.373
Academic curriculum covers in depth knowledge of Software engineering (SE).	.002	-.008	.133	.736	.009	.080	-.224	.222	-.025
Implementation(Practicals) of software engineering concepts was part of curriculum.	.056	.007	.847	.120	-.137	-.076	-.074	.116	-.128
Marks of implementation (practical) of SE added to marks of final degree.	.059	-.040	.839	-.018	.161	-.063	.080	.051	.164
SE syllabus covers basic knowledge of software testing.	-.320	.059	.147	-.243	.686	.009	.027	-.045	.061
Academic curriculum of SE meets the requirements of present s/w industry	-.070	.132	.099	.087	-.124	-.058	.076	.840	.140
ST was a separate subject with focus on manual testing in academic curriculum.	-.136	.001	-.174	.314	.660	-.351	-.007	-.020	-.020
ST syllabus provided knowledge of black box testing.	.264	.040	-.304	.005	.082	-.086	.708	.223	-.056
ST syllabus provides knowledge of white box testing.	.294	.092	.207	-.001	-.026	.020	.528	-.026	-.046
In depth knowledge of complete software testing was part of curriculum.	-.248	-.198	.274	.336	.283	-.131	-.183	.321	-.007
Final project training is actually done at s/w industry in testing domain	.801	.119	.054	-.057	-.250	.089	.002	.001	.059
Project classes and practical viva was under control of industry expert.	-.701	.272	-.063	-.084	.030	-.056	-.217	.095	.236
Implementation (Practical) of software testing strategies was part of curriculum.	.045	.889	-.126	-.188	-.002	.038	-.005	.144	-.028
SE curriculum covers architectural design of software.	-.251	-.044	-.035	.708	-.033	-.123	.345	-.162	.064
Regression testing is part of theory/software maintenance.	-.146	-.007	.129	-.124	.152	-.786	.210	-.102	-.098
Manual s/w testing concepts used in industry are same as you studied.	.505	.460	.068	.131	-.055	.272	.147	-.209	-.219
More emphasis on programming languages than SE in curriculum	.287	.236	.028	.130	-.122	.030	-.290	.084	.663
Live projects of ST were included in curriculum through industry collaboration .	.235	.186	-.080	-.168	-.010	.653	.204	-.196	-.005
Implementation (Practical) of software testing strategies should be part of curriculum.	-.883	-.129	.050	.057	.113	-.116	-.073	.067	-.035
Marks of implementation (practical's) of ST added to marks of final degree.	.395	.666	.128	.141	.033	.148	.200	-.131	-.021
Able to join software testing job directly after completion of academics.	-.019	.398	-.004	.043	-.122	-.081	-.118	-.040	-.708
ST syllabus provides knowledge of object oriented testing.	-.696	-.109	-.055	.156	.253	.122	-.075	-.071	-.261
SE concepts used in industry are same as you studied.	-.274	-.270	.128	-.003	.442	.040	.047	.573	-.185
For manual testing job, training is required from industry /private institute.	.832	.161	-.010	-.067	-.044	.128	.222	-.035	.078
In house training is required for manual testing .	.847	.113	.016	-.103	.012	.166	.001	-.138	.118

Extraction Method: Principal Component Analysis. Rotation Method: Varimax with Kaiser Normalization.

Factor 2: In factor 2, four variables found to be high factor loading i.e. .808(Demo version of automated testing tools were not used for practical of software testing), .822(Live projects of automated ST were not included in curriculum through industry collaboration), .858(Licensed tools of automated testing with full functionality were not part of curriculum), and .772(Final project training is actually done at s/w industry in testing domain). So these four variables can be clubbed and factor may be named as "Lack of practical of automated s/w testing in curriculum".

Factor 3: In factor 3, two variables found to be high factor loading i.e. .675(Implementation (Practical) of software testing strategies was part of curriculum), and .795(Marks of implementation (practical's) of ST added to marks of final degree). So these two variables can be clubbed and factor may be named as "Software testing syllabus includes practical of testing techniques and marks are added to degree".

Factor 4: In factor 4, two variables found to be high factor loading i.e. .825(Academic curriculum provides basic knowledge of software testing), and .718(In depth knowledge of software testing was part of curriculum). So these two variables can be clubbed and factor may be named as "Software testing has theoretical foundation in curriculum".

Table-5: Total Variance Explained

Component	Initial Eigenvalues			Extraction Sums of Squared Loadings			Rotation Sums of Squared Loadings		
	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %
1	7.307	29.229	29.229	7.307	29.229	29.229	4.598	18.391	18.391
2	2.562	10.250	39.479	2.562	10.250	39.479	2.989	11.957	30.348
3	2.187	8.749	48.228	2.187	8.749	48.228	2.225	8.899	39.247
4	1.474	5.895	54.122	1.474	5.895	54.122	2.177	8.710	47.957
5	1.253	5.011	59.133	1.253	5.011	59.133	1.952	7.810	55.767
6	1.177	4.707	63.840	1.177	4.707	63.840	1.789	7.157	62.924
7	1.076	4.305	68.146	1.076	4.305	68.146	1.305	5.222	68.146
8	.923	3.693	71.839						
9	.874	3.495	75.334						
10	.728	2.910	78.244						
11	.669	2.676	80.920						
12	.616	2.463	83.383						
13	.601	2.405	85.789						
14	.500	1.998	87.787						
15	.473	1.890	89.677						
16	.423	1.692	91.369						
17	.353	1.414	92.783						
18	.337	1.350	94.133						
19	.308	1.232	95.365						
20	.295	1.179	96.544						
21	.258	1.033	97.577						
22	.213	.850	98.427						
23	.172	.689	99.116						
24	.141	.563	99.679						
25	.080	.321	100.000						

Extraction Method: Principal Component Analysis.

Factor 5: In factor 5, three variables found to be high factor loading i.e. ..624(Automation testing was not separate subject in curriculum), .789(Automation testing was introductory part of curriculum), and .554(In depth knowledge of automated testing tools was not part of curriculum). So these three variables can be clubbed and factor may be named as "Lack of theory of automated s/w testing in curriculum".

Factor 6: In factor 6, two variables found to be high factor loading i.e. .804 (ST syllabus provides knowledge of various testing techniques), and .553(ST syllabus provides knowledge of object oriented testing). So these two variables can be clubbed and factor may be named as "Software testing syllabus includes theory of testing techniques".

Factor 7: In factor 7, one variable found to be high factor loading i.e. .628(Academic curriculum doesn't meet the requirements of present s/w industry). So this variable can be named as "Academic curriculum of CS/IT courses doesn't meet the requirement of industry".

4. CONCLUSION AND RECOMMENDATION

The findings presented in this paper were drawn from the responses of two surveys: 107 respondents for manual testing and 104 respondents for automated testing from the Indian IT industry. We believe that the findings are general in nature with actions required from academia and industry.

For manual testing nine factors are identified viz.

There is no gap between theoretical and actual implementation of manual s/w testing;
Software testing syllabus includes practical of manual testing techniques and marks are added to degree;
Software engineering practical are not adequate in curriculum but marks are added to degree;
Software engineering has theoretical foundation in curriculum; Software testing has theoretical foundation in curriculum;
Live projects of ST are not included in curriculum through industry collaboration;
Software testing syllabus includes theory of manual testing techniques;

Table-6: Rotated Component Matrix (Rotation converged in 11 iterations)

	Component						
	1	2	3	4	5	6	7
Academic curriculum meets the requirements of present s/w industry.	-.202	.014	.278	.103	.022	-.106	.628
Academic curriculum provides basic knowledge of software testing (ST).	-.254	.034	-.212	.825	.094	.011	-.078
In depth knowledge of software testing was part of curriculum.	-.172	.195	-.116	.718	.199	-.022	.244
ST should be included as a separate subject in academic curriculum.	.201	-.191	-.522	.216	-.125	.244	.380
Automation testing was separate subject in curriculum.	.115	-.166	-.034	.408	.624	-.126	-.071
Automation testing should be part of software testing.	.592	-.038	-.291	-.182	.032	.051	.447
Implementation (Practical) of software testing strategies was part of curriculum.	-.293	.042	.675	-.185	.105	-.038	.024
Marks of implementation (practical's) of ST added to marks of final degree.	-.094	.097	.795	-.028	.066	.105	.205
Demo version of automated testing tools were used for practical of software testing.	.080	.808	.033	-.132	.070	.356	.060
Automation testing was introductory part of curriculum.	-.291	.229	.069	-.090	.789	.152	.130
Automation testing was part of software testing.	-.565	.024	.301	.323	.072	.159	-.139
ST syllabus provides knowledge of various testing techniques.	.247	-.015	-.016	-.035	-.042	.804	-.012
Automation testing tools learned by private training institute.	.688	-.097	-.278	-.223	-.170	.174	-.177
Automation testing was part of software engineering subject.	-.185	.133	.447	.484	-.074	-.340	.114
In depth knowledge of automated testing tools was part of curriculum.	-.296	.188	.248	.095	.554	-.147	-.230
Implementation (Practical) of software testing strategies should be part of curriculum.	-.682	.074	.245	.096	.184	-.068	.080
Project classes and practical viva must under control of industry expert.	-.644	.183	.184	.309	.230	-.311	.105
Live projects of automated ST were included in curriculum through industry collaboration.	-.099	.822	.096	.226	.141	-.053	-.092
Automation testing tools learned by in house training by company.	.690	-.077	-.140	-.080	.200	.308	-.019
Licensed tools of automated testing with full functionality were part of curriculum.	-.078	.858	-.005	.053	-.028	-.113	.077
Automation testing tools learned by self-study /google /you tube.	.694	-.188	.057	-.090	-.072	.118	-.388
Final project training is actually done at s/w industry in testing domain	-.294	.772	.237	.100	.104	-.245	-.030
ST syllabus provides knowledge of object oriented testing.	.543	-.046	.052	.005	-.262	.553	-.165
Industries provide in house training for self-developed testing tools.	.806	-.015	-.020	.030	-.148	.109	.066
All automation testing tools could not be included in curriculum.	.115	-.122	-.149	-.286	-.502	.306	-.318

Extraction Method: Principal Component Analysis. Rotation Method: Varimax with Kaiser Normalization.

Academic curriculum of SE meets the requirements of present s/w industry and SE concepts used in industry are same as studied;
More emphasizes on programming language.
The conclusion of all these is that there is no gap between theory and practices of manual testing.

For automated testing, seven factors are identified viz.

There is gap between theoretical and actual implementation of automation s/w testing and automated testing tools are learned during job not in academic;
Lack of practical of automated s/w testing in curriculum;
Software testing syllabus includes practical of testing techniques and marks are added to degree;
Software testing has theoretical foundation in curriculum;
Lack of theory of automated s/w testing in curriculum;
Software testing syllabus includes theory of testing techniques;
Academic curriculum of CS/IT courses doesn't meet the requirement of industry.
The conclusion of all these is that there is gap between theory and practices of automated testing.

It is recommended that collaboration between the industry and academia including faculty training and curriculum development is crucial to bridge this gap. Students must be exposed to testing tools that are standard in the software industry. They need to acquire more practical testing experience and work on real-life projects that will permit them to acquire technical, soft and IT offshore outsourcing skills.

5. REFERENCES

[1] S.P. Ng, T. Murnane, K. Reed, D. Grant, T.Y., Chen A, 2004. Preliminary Survey on Software Testing Practices in Australia, Proceedings of the 2004 Australian Software Engineering Conference (ASWEC'04).

[2] Emelie Engstr¨om and Per Runeson, 2010. A Qualitative Survey of Regression Testing Practices, LNCS 6156, Springer-Verlag pp. 3–16.

[3] Vidya Kulkarni, Christelle Scharff, et. al 2010. From Student to Software Engineer in the Indian IT Industry: A Survey of Training, 23rd IEEE Conference on Software Engineering Education and Training.

[4] Causevic, A., Sundmark, D., Punnekkat, S, 2010. An industrial survey on contemporary aspects of software testing, Proceedings of the 3rd International Conference on Software Testing Verification and Validation, pp. 393–401.

[5] Rooksby, J., Rouncefield, M., Sommerville,I, 2009. Testing in the wild: The social and organisational dimensions of real world practice, Computer Supported Cooperative Work (CSCW) 18(5), pp 559–580.

[6] Grindal, M., Offutt, J., Mellin, J, 2006. On the testing maturity of software producing organizations. In: Testing: Academia & Industry Conference-Practice And Research Techniques, TAIC/PART.

[7] Murnane and K. Reed, 2001. On the Effectiveness of Mutation Analysis as a Black-box Testing Technique, Proceedings of the 2001 Australian Software Engineering Conference, Canberra, Australia, pp 12-20.

[8] Paul C Jorgensen, 2013. Software Testing: A craftsman's Approach. CRC Press.

[9] R. S. Pressman, 2004. Software Engineering: A Practitioners Approach, McGraw-Hill, International Edition.

[10] K. Reed, 2002. Testing, testing, testing, Australian Computer Society Information Age, pp 56–58.

[11] L. Groves, R. Nickson, G. Reeves, et. al 2000. A Survey of Software Practices in the New Zealand Software Industry", Proceedings of the Australian Software Engineering Conference, pp 189-101.

[12] IEEE: IEEE standard for software test documentation. IEEE Std(829-1983, Revision) (1998)