The Reality of Software Reverse Engineering Education in the Jordanian Universities and How to improve it

Rawan AbuLail
Computer Science Department
Philadelphia University
Amman, Jordan
rawanabulail@yahoo.com

Mohammad Shkoukani
Computer Information Systems Department
Applied Science University
Amman, Jordan 11931-166
m.shkokani@asu.edu.jo

Abstract—**In order to apply reverse engineering in software industry in an efficient and effective way there should be a high awareness in software reverse engineering concepts, approaches, and tools for employees in the software industry. University is considered as the main resource for supplying employees with awareness and knowledge in any field.**

Despite of software reverse engineering importance, its adoption in the educational institutions especially in the universities is still limited.

This paper has three objectives, First, it highlights the importance of software reverse engineering education to the students; second, it determines whether there is software reverse engineering awareness in the Jordanian universities; the third objective of this paper is to develop an approach in order to increases software reverse engineering awareness in Jordanian universities.

The result of the study (56) respondents of software engineering department's educators indicated that unfortunately, there is no software reverse engineering awareness in Jordanian universities. Another indication was that the developed approach would have a significant impact on improving the education of software reverse engineering in Jordanian universities.

Keywords- Software reverse engineering, Software reverse engineering education, Software reverse engineering awareness

I. INTRODUCTION

One of the most important processes in software engineering is reverse engineering. It uses the life cycle of system development in backwards manner in order to recapture the representation of a system by analyzing structure, operations and functions of that system. The importance of reverse engineering has been emerged because most of old systems have no written documentations and in some cases the developer for a certain system is no longer working in the developing company [1, 2].

The only accurate resource that software developer and maintainers can use is the source code, so they can use it directly for high-level structural information recovering by developing new techniques and tools to deal of that issue. The input of reverse engineering process could be either software source code or the executable code; we can use the executable code when the source code is lost [31, 32].

The adoption of software reverse engineering concept is still limited especially in the academic institutions and it needs more effort from researchers. Although the software reverse engineering researches and published papers have a good results and contributions but we still need more empirical studies in this field in order to make comparisons between the theory techniques and asses them, and on the other hand to introduce and develop some new techniques [9].

Based on the investigation of software industry, the universities failed to prepare students for real world because there are many skills that are ignored [6]. So, universities should pay more attention to software reverse engineering in order to provide software industries with well-educated software engineering graduates with high competence in reverse engineering field. The researchers are attempting to answer: Is there a software reverse engineering awareness in the Jordanian universities; how can we increases software reverse engineering awareness in the Jordanian universities.

II. PROBLEM STATEMENT

This paper investigates the reality of software reverse engineering awareness in software engineering departments in Jordanian universities and how to improve it.

The following questions can better help clarifying the problem identified in this paper:

1. Is there software reverse engineering awareness in Jordanian universities?
2. What are the most significant factors that impact on software reverse engineering awareness in Jordanian universities?

III. STATEMENT OF THE RESEARCH OBJECTIVES

The paper's objectives are to build and test a suggested model that supports software reverse engineering awareness in Jordanian universities. In addition to developing an approach for teaching software reverse engineering in Jordanian universities, the proposed approach will increase reverse engineering awareness in software engineering departments in Jordanian universities depending on four dimensions that were defined in the research model leading to increase software reverse engineering awareness.

IV. RESEARCH HYPOTHESES

The researchers have set four hypotheses:

H01: There is no software reverse engineering awareness in Jordanian universities toward scientific research budget.

H02: There is no software reverse engineering awareness in Jordanian universities toward library's capabilities and budget.

H03: There is no software reverse engineering awareness in Jordanian universities toward academic staff capabilities and properties.

H04: There is no software reverse engineering awareness in Jordanian universities toward software reverse engineering infrastructure and tools.

V. SUGGESTED MODEL

According to some previous studies in software reverse engineering education [8, 10, 11, 12] the researchers suggested the following model which is consisted of two types of variables, the independent variable which is include scientific research budget, library's capabilities and budget, academic staff capabilities and properties, and software reverse engineering infrastructure and tools. The dependent variable which is software reverse engineering awareness in Jordanian universities, as shown in "fig 1" below.

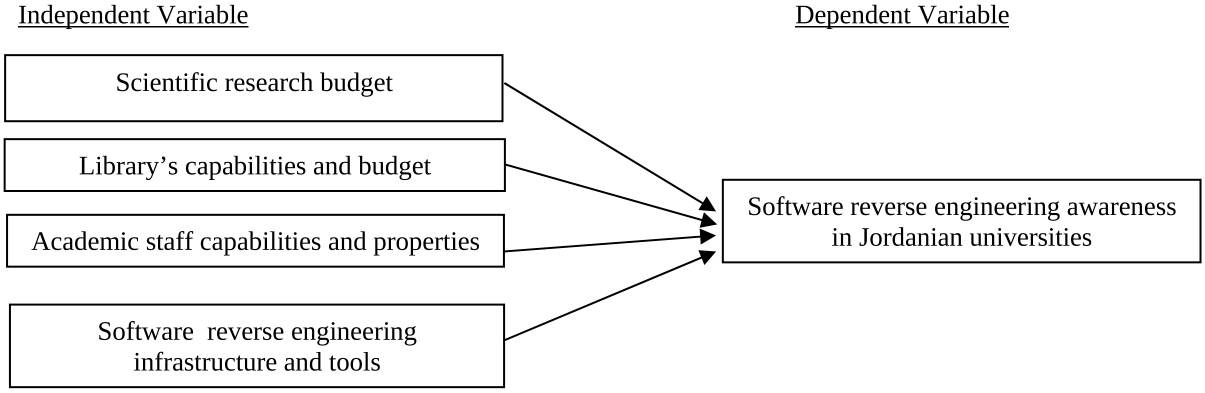
Independent Variable
Dependent Variable
Scientific research budget
Library's capabilities and budget
Academic staff capabilities and properties
Software reverse engineering infrastructure and tools
Software reverse engineering awareness in Jordanian universities

Figure 1: Research Model

VI. STUDY POPULATION

To examine the research problem and the questions raised in that respect and in order to fulfill the objectives of this paper, the researchers have chosen to conduct their research on the software engineering departments in Jordanian universities.

The researchers have made a survey for all educators in the software engineering departments in Jordanian universities (both state and private universities), 75 questionnaires were distributed to all educators in software engineering departments, 64 questionnaires were returned, and 56 were valid responses.

VII. RELIABILITY OF THE INSTRUMENT

SPSS version 19 was used by the researchers in order to analyze the collected data and to infer the appropriate conclusion [28].

Cronbach alpha was used to test the internal reliability of the measurement instrument. In this paper 0.60 or higher is considered acceptance [29].

Table 1 shows the Cronbach Alpha values which ranged from .821 to .933, so establishing the reliability of the survey. It is obvious that all values of alpha are higher than .60. This indicates that the items are highly correlated and highly consistent.

TABLE 1. CRONBACH'S ALPHA FOR EACH SCALE

No.	Variables	Cronbach alpha
1.	Software reverse engineering awareness in Jordanian universities	.908
2.	Scientific research Budget	.821
3.	library’s capabilities and budget	.933
4.	Academic staff capabilities and properties	.911
5.	Software reverse engineering infrastructure and tools	.903

VIII. LITERATURE REVIEW

The main objective of software reverse engineering as a discipline in software engineering is extracting high-level descriptions of systems using its source code in order to produce documents that can be used as supplements and in some situations it could be used for replacing outdated or missing documentations [4, 5].

It can be defined also as the process of identifying the components of a given system and then find the relationships between those components in order to create another form of representation at a higher levels of abstractions. On one hand software reverse engineering process doesn’t involve creating a new systems or making changes to the system because it doesn’t affect the program itself, on the other hand there are several software engineering activities affect the system itself, like software testing, integration, reuse, maintenance and quality assurance, these examples of software processes affect the systems itself because they are involving existing systems [8]. One of the key aspects of these examples is identifying the system’s components and understanding the relationships between them [6, 9].

Software reverse engineering can support the software quality improvements on many levels of abstractions , the first level is the source –code, and higher levels like the requirements and design levels, by providing varying representations and alternative views of the systems. These improvements can eliminate the redundancies in the system in an effective way to avoid the quality deterioration in the maintenance operations [13, 33].

A. Importance and advantages of software reverse engineering education

The importance of reverse engineering comes from the advantages that the student gains from learning reverse engineering some of them are as the followings: deeper and better understanding of the systems which is the foremost and first advantage of teaching the concepts of reverse engineering. The advantages of teaching reverse engineering can be listed as follows:

The most important advantage of reverse engineering concepts education is the deeper and better understanding of the systems. That means that when students have the practice skill of using reverse engineering techniques it will improve their abilities to understand a given system efficiently and quickly. This skill has very important economic effects especially when calling software engineers to solve problems related to software evolution. Reverse engineering plays an important role in systems understanding and from there higher level abstractions recovery. No doubt that reverse engineering is time consuming, difficult and painstaking on one hand and on the other hand it is as well very informative [2, 13].

There are many industries that changing in a rapid way, one of them is the software industry. The skills of designing and programming can be improved by employing reverse engineering for students in projects and assignments. Educators must enhance students to self-analyze their programs and systems /design by using reverse engineering techniques to indicate limitations and eliminate them. Another way that educators can use is to let their students analyze their peers' assignments. One of the problems that software engineering graduates face in industry is the differences between the actual programs/ systems in industry and the programs/systems they deal with in education. The programs which written by students are very straightforward and small compared to systems and programs in industry. And because of that there was a conducting for a very unique research [16], which is focusing on using software with open source as a classroom material, instead of implementing and designing new systems. An open source system is given for students and then asking them to modify it. Many companies realized the importance of reverse engineering skill in cost cutting of legacy systems maintenance and in minimizing the software evolution risk [30].There is a pressure from software industry to develop new and better methodologies for reverse engineering. Software engineering researches must focus on software reengineering and maintenance. Software literature must cover legacy system as an essential part in it [6, 15, 16].

Educators have a responsibility to improve many skills for students one of them is the problem solving skill, when students deal with software integration and legacy systems issues they must have a strong problem solving and analytical skills. We don't require building new system from scratch every time. In some cases existing systems requiring modification and maintenance, most of fresh software engineering graduates have no experience in software reengineering and reverse engineering and because of that they are unable to work on such projects. Software reverse engineering education for graduates has opened an exciting and new avenue for them in industry in their career. While they working with legacy systems they can improve skills which give them an edge. They must take into account the changing trend in software industry [5, 33].

In most universities software reverse engineering is offered at advanced levels not at undergraduate levels. It's very important to consider reverse engineering in undergraduate software engineering curricula an essential part at all universities. The syllabus of traditional software engineering is based on forward engineering. That is the reason why most of fresh software engineering graduates facing difficulties in working with legacy systems and they hardly interested in working with it. Another reason for those difficulties is the lack of essential skills needed to deal with legacy systems. We can help the fresh graduates of software engineering by educating the students reverse engineering principles to help them gaining the analytical skills which they need to deal with the legacy systems [17].

B. Software reverse engineering adoption in academic universities

The adoption of reverse engineering in research and education is still limited and requires to be favored by means. First the discipline of reverse engineering must be educated at different levels. Second there are quite solid research papers and publications in reverse engineering which contain solid validation; also there is a need for empirical studies covering the comparison issues for reverse engineering techniques, in order to assess the usability of tools and their usefulness, and to understand the suitable circumstances in which we can apply particular techniques [25]. The empirical studies in reverse engineering playing roles in assessing existing theories of reverse engineering and in developing new theories dealing with program comprehension in order to make sense of conclusions and data [14].

One of the most important issues for the next decade that we must take it into account is teaching students the software evolution and maintenance because it is neglected in most curricula in software engineering, computer engineering and computer science departments while these departments just concentrate on teaching students how to construct software from scratch [2, 26].

There are concepts like efficiency, consistency, architecture, completeness, robustness or abstraction should be taught in two perspectives which are software analysis and software design. Computer science programs are now establishing many courses related to the software architecture while there is a lack in many courses which related to software migration, reverse engineering, software reengineering, software evolution, or program understanding [3, 30].

In software engineering education the trend is to teach students how to develop small programs and new systems from scratch, not to teach them how to change and understand existing systems and large ones. There is a critical need to emphasis on the issues related to software change and evolution for both graduate and undergraduate programs in software engineering by teaching students the theory and practice of software reengineering, comprehension and maintenance [3, 7, 13, 27].

C. Previous studies of software reverse engineering education

There are many studies highlighting the importance of reverse engineering education for students, some of these studies as follows:

Ahmad K. Ghafarian was one of the most important researchers in this topic, he has a long experience in the process of incorporating the techniques of reverse engineering on software engineering education in order to enhance the productivity and the quality of students team projects. The approach which he used in his experiments was to make the instructors provide their students existing working software, then the instructors asked their students to evaluate the software using the knowledge from the behavior of the system and the domain of that application in order to define the interaction pattern that reflected by the task scenarios and then define the services provided by the system to its environment. Consequently developing the architectural design for the software system using the extracted patterns to rebuild an improved version for the software. After conducting this experiment the responses of the overall students indicate that although the task was challenging it was so rewarding experience [8].

Missouri-Rolla University conducted an interesting research about reverse engineering techniques and the other goal for that study was to encourage students to study and learn the real world products, after conducting the study the results was that 77% of the students thought that the concepts taught during courses and lectures can be reinforced by introducing reverse engineering methodologies, and 82% of them would incorporated in future subjects and courses especially courses related to design topic[18].

There was another study concerning about the training of software engineering students, the idea in this study is to let students work on actual and real commercial projects, the training period for the students can be from six months up to two years, for some universities this training period may be impossible, and because of that there was another idea which was developing systems and software by simulating the conditions of real life. The simulation can be done by letting software engineering students cooperate with students in other departments and develop ‘serious’ systems and software for them. This practice would be good in developing systems and software according to the prospective users' requirements. The study also shows the importance of designing the software engineering curriculum which must equip software engineering graduates having an up to date software engineering technical skills. There is a need for software engineering graduates who are able to work with legacy systems, and because of that software engineering curriculum must include software reverse engineering / reengineering courses. The educators of software engineering must always improve the methodologies of teaching software engineering courses and must always make the materials of courses up to date upon the software engineering industry needs and requirements. Finally the study highlighting the importance of designing the programs and curriculums of software engineering according to standards guidelines and getting approvals from some third parties like software engineering body of knowledge (SWEBOK) [19, 20, 21, 22, 23].

An Indian study found that most of Indian universities focus in their software engineering syllabus on theoretical aspects and there is a lack of focus on software engineering applications, knowledge and skills. Recent development, best practices, practical aspects are not included in the syllabus. And the study found that students don't have a chance to employ the skills and knowledge they learned in real applications and systems. Software engineering industry in India faces a problem in finding well qualified graduates of software engineering who can fulfill the software engineering industry needs. The study also shows that there is a need for qualified software engineering educators who have a considerable experience in the development of systems and software, not just an experience in teaching software engineering courses, in order to discuss problems related to the development of real project, the intricacies of building real systems and software development practices in a convincing manner, in addition those educators can also talk about the theoretical aspects in software engineering.

Finally the study found that one main hindrances of reforming the software engineering education is the lack of proper software engineering infrastructure. And then the study emphasized on the importance of increasing the number of qualified institutions that offer software engineering programs, taking into account the curriculum design, the facilities and the quality of teaching. These facilities must include qualified and well trained software engineering educators, high quality study material and access to software engineering tools [11, 24].

IX. DATA ANALYSIS AND RESULTS

A. *Population Description*

This section describes the population which is software engineering educators in Jordanian universities through the general characteristics of the respondents in term of university’s sector, gender, age, scientific qualification, academic rank, general major, specialist major, teaching experience, teaching experience in software engineering department, number of specialized courses in RE, number of published researches in RE, and research interests as shown in table 2 below.

TABLE 2 POPULATION DESCRIPTIONS

Variable	**Category**	**frequency**	**Percent%**
Sector	Government	16	28.6
	Private	40	71.4
Variable	**Category**	**frequency**	**Percent%**
Gender	Male	41	73.2
	Female	15	26.8
Variable	**Category**	**frequency**	**Percent%**
Age	Under 30	3	5.4
	31-40	22	39.3
	41-50	22	39.3
	Over 50	9	16.1
Variable	**Category**	**frequency**	**Percent%**
Scientific qualification	Master	8	14.3
	PhD	48	85.7
Variable	**Category**	**frequency**	**Percent%**
Academic rank	Master	8	14.3
	Assistant professor	41	73.2
	Associated professor	7	12.5
	Professor	0	0.0
Variable	**Category**	**frequency**	**Percent%**
General major	Software engineering	15	26.8
	Else	41	73.2
Variable	**Category**	**frequency**	**Percent%**
Specialist major	Reverse engineering	1	1.8
	Else	55	98.2
Variable	**Category**	**frequency**	**Percent%**
Teaching experience	1-5	23	41.1
	6-10	12	21.4
	More than 11	21	37.5
Variable	**Category**	**frequency**	**Percent%**
Teaching experience in software engineering department	1-3	36	64.3
	4-7	15	26.8
	More than 8	5	8.9
Variable	**Category**	**frequency**	**Percent%**
Number of specialized courses in RE	0	49	87.5
	1	4	7.1
	2	3	5.4
Variable	**Category**	**frequency**	**Percent%**

Number of published researches in RE	0	43	76.8
	1	4	7.1
	2	4	7.1
	3	2	3.6
	4	2	3.6
	5	1	1.8
Variable	**Category**	**frequency**	**Percent%**
Research interests	Reverse engineering	14	25.0
	Else	42	75.0

B. Central Tendency Measures

There are many measures of central tendency such as mean, median, mode, and standard deviations [29]. Mean and standard deviations were used to describe attitudes toward the following questions:

TABLE 3: SCIENTIFIC RESEARCH BUDGET RESULTS

No.	Scientific research budget	Mean	Std. Deviation
1	Deanship of academic research supports faculty members to participate in scientific conferences and research papers publications.	3.64	.841
2	Deanship of academic research encourages and supports the software engineering department to hold conferences and seminars of their own.	3.32	.917
3	Deanship of academic research supports outstanding students whom have developed their own software projects.	2.84	1.092
4	Deanship of academic research supports faculty members by covering the costs of participation in scientific conferences.	3.16	.890
5	Deanship of academic research rewards the best researcher at the software engineering department.	2.43	1.076
6	Deanship of academic research supports the printing and publication of RE books.	2.59	1.108
7	Deanship of academic research encourages RE workshops.	2.46	1.078
8	The budget of deanship of academic research is sufficient to support all research in the software engineering department.	2.86	.943

Table 3 shows the mean ratings of scientific research budget. Based on the responses, most of the mean ratings are less than 3 on the five-point scale.

As can be seen from the table “Deanship of academic research rewards the best researcher at the software engineering department“ has the lowest mean score of (2. 43).

TABLE 4: LIBRARY’S CAPABILITIES AND BUDGET RESULTS

No.	Library’s capabilities and budget	Mean	Std. Deviation
1	There are many RE books and references at the university's library.	2.66	1.180
2	There are enough copies of RE books commensurate with the number of students in software engineering department.	2.80	1.212
3	Availability of RE text books in university’s library.	3.07	1.042
4	Availability of an E-Library in the university.	3.20	1.151
5	University subscribes to some scientific journals in RE areas.	3.00	1.112
6	RE books and references are up to date.	2.80	1.212
7	The Library knows the shortcomings of RE books and references by teachers and students regularly.	2.75	1.195
8	The Library addresses the lack of RE books rapidly.	2.54	1.279
9	University Library is an integrated and supportive source of teaching and research in RE field.	2.55	1.320

As shown in table 4 the perception of library’s capabilities and budget was investigated by asking respondents to rate each of the nine statements. Perceptions of library capabilities and budget were measured on a five-point rating scale. Most of the rating scores were less than 3, the lowest mean score (2.54) which was “The Library addresses the lack of RE books rapidly”.

TABLE 5: ACADEMIC STAFF CAPABILITIES AND PROPERTIES RESULTS

No.	Academic staff capabilities and properties	Mean	Std. Deviation
1	Faculty members participate in scientific conferences and seminars in RE.	2.88	1.080
2	Faculty members subscribe to scientific journals such as IEEE and ACM.	3.30	.971
3	The faculty members review RE courses quarterly and continuously to remain updated with the changes in the field in order to update the RE course curriculums.	3.29	.967
4	The faculty members help and encourage their students in R.E fields.	2.89	1.107
5	The faculty members encourage and urge their students on graduation projects into R.E area.	2.77	1.079
6	The faculty members follow up their students from first year to the graduation year to find their weaknesses in RE.	2.77	1.079
7	The faculty members monitor the progress of students during the training period and measure the benefit of practical training in RE.	2.86	1.135
8	The faculty members help organizing seminars in novel R.E areas.	2.63	1.169

A seen from table 5 the participants’ opinions about Academic staff capabilities and properties were rated on five-point scale. Based on the responses, only “Faculty members subscribe to scientific journals such as IEEE and ACM” and “The faculty members review RE courses quarterly and continuously to remain updated with the changes in the field in order to update the RE course curriculums” are with a mean rating greater than 3 on the five-point scale. All other statements were with a mean rating less than 3. “The faculty members help organizing seminars in novel R.E areas” had the lowest mean score (2.63).

TABLE 6: SOFTWARE REVERES ENGINEERING INFRASTRUCTURE AND TOOLS RESULTS

No.	Software reveres engineering infrastructure and tools	Mean	Std. Deviation
1	Number of laboratories is sufficient and commensurate with student number.	3.59	1.041
2	There are enough modern computers in laboratories that increase the effectiveness of the educational process.	3.21	.967
3	There is some practical RE material that is taught in laboratories.	2.88	1.113
4	For each class room based module in RE, there is an allocated number of laboratory hours per week.	2.71	1.187
5	There is sufficient R.E material that is taught in the software engineering department.	2.66	1.164
6	RE material is part of software engineering material and is not taught separately.	3.11	1.090
7	Lecturers change the content of RE courses quarterly and continuously to remain up to date with changes and developments in the field.	2.91	1.133
8	Lecturers adopt R.E textbooks given the requirements of the course and are not relying on what is available in the library.	3.16	.987
9	There are equipments that support the delivery approach in each class room such as Data show and Internet access.	3.46	.873
10	RE course material is logically segmented to be covered for all levels of software engineering students.	2.43	1.093
11	Specialized software tools are adopted for teaching RE courses in laboratories.	2.29	1.155

The Participants' opinions about R.E Infrastructure and tools were rated on five-point scale. Based on the responses, most of the mean ratings are less than 3 on the five-point scale. As seen from table 6, "Number of laboratories is sufficient and commensurate with students' number" had the greatest mean score (3.59). "Specialized software tools are adopted for teaching RE courses in laboratories" had the lowest mean score (2.29).

C. Hypotheses Testing

Hypothesis (1):

Ho1: There is no software reverse engineering awareness in Jordanian universities toward scientific research budget.

Ha1: There is software reverse engineering awareness in Jordanian universities toward scientific research budget.

TABLE 7: TEST OF HYPOTHESIS (Ho1)

T calculated	T tabulated	T Sig.	Result of Ho1
- .981	2.925	.331	Accept

Referred to table 7, one sample T-test used at .05 significant level to test the hypothesis and it was found that (calculated T = - .981) is less than tabulated T, and Sig. T value is greater than 0.05. According to the decision rule: Accept Ho if calculated value is less than tabulated value and reject Ho if calculated value is greater than tabulated value. Therefore, Ho1 is accepted and Ha1 is rejected. This indicates that there is no software reverse engineering awareness in Jordanian universities toward scientific research budget.

Hypothesis (2):

Ho2: There is no software reverse engineering awareness in Jordanian universities toward library's capabilities and budget.

Ha2: There is software reverse engineering awareness in Jordanian universities toward library's capabilities and budget.

TABLE 8: TEST OF HYPOTHESIS (Ho2)

T calculated	T tabulated	T Sig.	Result of Ho2
- 1.413	2.925	.163	Accept

According to table 8, one sample T-test used at .05 significant level to test the hypothesis and it was found that (calculated T = - 1.413) is less than tabulated T, and Sig. T value is greater than 0.05. Therefore, Ho2 is accepted and Ha2 is rejected.

This indicates that there is no software reverse engineering awareness in Jordanian universities toward library's capabilities and budget.

Hypothesis (3):

Ho3: There is no software reverse engineering awareness in Jordanian universities toward academic staff capabilities and properties.

Ha3: There is software reverse engineering awareness in Jordanian universities toward academic staff capabilities and properties.

TABLE 9: TEST OF HYPOTHESIS (Ho3)

T calculated	T tabulated	T Sig.	Result of Ho3
- .693	2.925	.491	Accept

Referred to table 9, one sample T-test used at .05 significant level to test the hypothesis and it was found that (calculated T = - .693) is less than tabulated T, and Sig. T value is greater than 0.05. Therefore, Ho3 is accepted and Ha3 is rejected. This indicates that there is no software reverse engineering awareness in Jordanian universities toward academic staff capabilities and properties.

Hypothesis (4):

Ho4: There is no software reverse engineering awareness in Jordanian universities toward reverse engineering infrastructure and tools.

Ha4: There is software reverse engineering awareness in Jordanian universities toward reverse engineering infrastructure and tools.

TABLE 10: TEST OF HYPOTHESIS (Ho4)

T calculated	T tabulated	T Sig.	Result of Ho3
- .522	2.925	.604	Accept

Referred to table 10, one sample T-test used at .05 significant level to test the hypothesis and it was found that (calculated T = - .522) is less than tabulated T, and Sig. T value is greater than 0.05. Therefore, Ho4 is accepted and Ha4 is rejected. This indicates that there is no software reverse engineering awareness in Jordanian universities toward reverse engineering infrastructure and tools.

X. CONCLUSIONS & RECOMMENDATIONS

A. Conclusions

The study shows that there is no software reverse engineering awareness in Jordanian universities, according to the following results:

- There is no software reverse engineering awareness in Jordanian universities toward scientific research budget because most of deanships of academic research in Jordanian universities do not support outstanding students whom have developed their own software projects, do not reward the best researcher at the software engineering department, do not support printing and publication of software reverse engineering books, and do not encourage software reverse engineering workshops.
- There is no software reverse engineering awareness in Jordanian universities toward library's capabilities and budget, since there are not many software reverse engineering books and references at the university's library, also there are not enough copies of software reverse engineering books that commensurate with the number of students in the software engineering department, in addition software reverse engineering books and

references are not up to date, the respondents added that library does not address the lack of software reverse engineering books rapidly and it cannot be considered as an integrated and supportive source of teaching and research in software reverse engineering field.

- There is no software reverse engineering awareness in Jordanian universities toward academic staff capabilities and properties because few of the faculty members participate in scientific conferences and seminars in software reverse engineering, the faculty members do not help or encourage their students in software reverse engineering fields, the also do not encourage their students on graduation projects into software reverse engineering areas, in addition they do not follow up with their students from first year to the graduation year to find their weaknesses in software reverse engineering, they do not help organizing seminars in novel reverse engineering areas. This result agrees with study [11] which showed that there is a lack of qualified software engineering teachers.

In addition it was found from demographic information that only (26.8%) of the respondents their general major was software engineering, and there is only one respondent out of all respondents whose specialist major is reverse engineering with percentage (1.8 %) which is a very little percentage, and the most respondents do not have any specialized courses in reverse engineering which represent (87.5%), also the majority of respondents (76.8%) do not have any reverse engineering publication, finally only (25.0 %) of respondents have reverse engineering research interests.

- There is no software reverse engineering awareness in Jordanian universities toward software reverse engineering infrastructure and tools, this result came from many reasons such as only few of the universities adopted specialized software tools for teaching software reverse engineering courses, software reverse engineering course material is not logically segmented to be covered for all levels of software engineering students, lecturers do not change the content of software reverse engineering courses quarterly and continuously to remain up to date with changes in the field, there is not sufficient software reverse engineering material that is taught in laboratories in software engineering department, this result agrees with study [19] which showed that few software engineering programs cover the subject of software change and evolution. Fewer courses cover some aspects of software reengineering. There are few practical software reverse engineering materials that are taught in laboratories, this result also agrees with [11] which showed that software engineering course syllabus in most of the Indian universities focuses on theoretical aspects and very little on application of software engineering skills and knowledge.

B. Recommendations

Based on the results the researcher recommends the following approach to be followed in reverse engineering education in software engineering departments in Jordanian universities to improve students' reverse engineering awareness in these universities. Researchers' approach consists of four dimensions:

First dimension of this approach related to the scientific research budget through the followings: deanship of academic research of the Jordanian universities should encourage and support software engineering department to hold conferences, seminars, and workshops in reverse engineering. It should also support and reward the outstanding students and best researchers in software engineering department, in addition it should also ensure and support the printing and publishing of reverse engineering books.

Second dimension related to the library's capabilities and budget. University's libraries should provide software engineering department with up to date reverse engineering books and references that are required and necessary for educational process, these books and references must be provide with enough copies that commensurate with the number of students in the department. These libraries should subscribe to some scientific journals in reverse engineering areas to increase efficiency of students in this field. They should address the lack of reverse engineering books quickly.

Third dimension related to the academic staff capabilities and properties which consist of the followings: faculty members in software engineering departments in Jordanian universities should participate in scientific conferences and seminars in reverse engineering. They should help and encourage their students in reverse engineering especially in their graduation projects. The faculty members should monitor the progress of their students during the training period and measure the benefit of practical training in reverse engineering. They should be also more aware and interested in reverse engineering field.

Fourth dimension of the approach which related to the reverse engineering infrastructure and tools included the followings: all of the practical reverse engineering materials should be taught in the laboratories, for each class room based module in reverse engineering there should be an allocated number of laboratory hours per week. Reverse engineering material should be taught separately and is not as part of software engineering material. The content of reverse engineering courses should be reviewed and changed quarterly and continuously to remain up to date with changes and developments in the field. These materials should be logically segmented to be covered for all levels of software engineering students. Finally specialized software tools should be adopted for teaching reverse engineering courses in laboratories.

ACKNOWLEDGMENTS

The authors are grateful to Philadelphia University, Amman, Jordan, for the financial support granted to cover the publication fees of this research article.

REFERENCES

[1] I. Rosziati and K. Yong, “ReSeT: Reverse Engineering System Requirements Tool”, proceedings of Word Academy of Science, Engineering and Technology, Vol. 32, pp. 272-275, 2008.

[2] H. M. Kienle and H. A. Müller, “Rigi-An environment for software reverse engineering, exploration, visualization, and redocumentation”, Science of Computer Programming, Volume 75 Issue 4, April, pp. 247-263, 2010.

[3] G. G. Richard, “A highly immersive approach to teaching reverse engineering”, Proceedings of the 2nd conference on Cyber security experimentation and test, 1-4, 2009.

[4] V. Jukka, “Reverse Engineering Framework Reuse Interfaces, Foundations of Software Engineering”, Proceedings of the 9th European software engineering conference held jointly with 11th ACM SIGSOFT international symposium on Foundations of software engineering, pp. 217-226, 2003.

[5] A. Muhammad, “Why teach reverse engineering?”, ACM SIGSOFT Software Engineering Notes, Vol. 30, Issue 4, pp. 1-4, Jul 2005.

[6] G. Canfora, M. Penta, and L. Cerulo, “Achievements and challenges in software reverse engineering”, Communications of the ACM, Volume 54 Issue 4,pp. 142-151, April 2011.

[7] M. El-Ramly, “Experience in teaching a software reengineering course”, ACM Special Interest Group on Software Engineering International Conference on Software Engineering Proceeding of the 28th international conference on Software engineering, pp 699-702, 2006.

[8] A. Ghafarian, “Reverse engineering technique to enhance software engineering education”, ACM,Annual Joint Conference Integrating Technology into Computer Science Education, Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education, pp. 354-356, 2005.

[9] G. Canfora and M. Penta, “New Frontiers of Reverse Engineering”, IEEE Computer Society , International Conference on Software Engineering, 2007 Future of Software Engineering, pp. 326-341, May 2007.

[10] J. Letizia, M. Sandro, “On the importance of dialogue with industry about software engineering education”,ACM, International Conference on Software Engineering, Proceedings of the 2006 international workshop on Summit on software engineering education, pp. 5-8, 2006.

[11] K. Garg and V. Varma, “People issues relating to software engineering education and training in India”, ACM, India Software Engineering Conference, Proceedings of the 1st conference on India software engineering conference, pp. 121-128, 2008.

[12] Ž. Mario, B. Ivana, and O. Marin, “Enhancing software engineering education: a creative approach”, ACM, International Conference on Software Engineering, Proceedings of the 2008 international workshop on Software Engineering in east and south Europe, pp. 51-58, 2008.

[13] F. Yang and H. Mei, “Development of software engineering: co-operative efforts from academia, government and industry”, ACM, International Conference on Software Engineering, Proceedings of the 28th international conference on Software engineering, pp. 2-11, 2006.

[14] Storey, “Theories, methods and tools in program comprehension: Past, present and future”, IEEE, Proceedings in the 13th International Workshop on Pogram Comprehension (IWPC), pp. 181-191, 2005.

[15] H. Ahmad, and H. Richard, “The small world of software reverse engineering”, IEEE,Proceedings of the 11th Working Conference on Reverse Engineering, pp. 8-12, 2004.

[16] C. David and K. Kim, “Teaching software design with open source software”, IEEE,Frontiers in Education Conference, Vol. 3, 2003.

[17] F. Anthony and K. Jeff, “Software Engineering: A Roadmap, The Future of Software Engineering”, ACM Press, 2000.

[18] S. Robert and M. Daniel, “The Touchy –Feely Side of Engineering Education: Bringing hands-on Experience to Classroom”, 35th ASEE Midwest Section Conference, 2000.

[19] B. Donald and G. James, ” Mengel Susan and Heinze Lloyd, Engineering Education Innovation With Software Engineering Projects”, ASEE/IEEE Proceedings of the 32nd of Frontiers in Education Conference, Vol. 3, pp. 13-16, 2002.

[20] H. Ellis and G. Hislop, “Techniques for Providing Software Engineering Education to Working Professionals”, ASEE/IEEE Proceedings of the 34th of Frontiers in Education Conference, Vol. 2, pp. 19-24, 2004.

[21] B. Brian, “A Student-Enacted Simulation Approach to Software Engineering Education”, IEEE Transactions on Education, Vol. 46, Issue 1, pp. 124-132, February 2003.

[22] H. James and T. Richard, “Developing the Software Engineering Team”, ACM, Proceedings of the 7th Australasian conference on Computing education, Vol. 42, pp. 203-210, 2005.

[23] K. Judith, U. Richard, H. Sara, and T. Raluca, “Teams in Software Engineering Education”, ASEE/IEEE Proceedings of the 32nd of Frontiers in Education Conference, Vol. 3, pp. 17-22, 2002.

[24] Mahanti, and Mahanti, “Software Engineering Education From Indian Perspective”, IEEE Computer Society, Proceedings of the 18th Conference on Software Engineering Education And Training, pp. 111-117, April 2005.

[25] Razmov, “Effective pedagogical principles and practices in teaching software engineering through projects”, Frontiers in education conference – global engineering: knowledge without borders, opportunities without passport, pp. 21-26, October 2007.

[26] Vaughn and Carver, “The Importance of Experience with industry in Software Engineering Education”, IEEE Computer Society, Proceedings of the 19th Conference on Software Engineering Education And Training Workshop (CSEETW), pp. 19-19, April 2006.

[27] C. Jill and B. Cathy, “Creating connections: bringing industry and educators together”,Conference on Information Technology Education (formerly CITC), Proceedings of the 6th conference on Information technology education, pp. 175-178, 2005.

[28] J. Marques de Sa, “Applied Statistics Using SPSS, STATISTICAL, MATLAB, and R”, 2nd Edition, Springer, 2007.

[29] Saunders, Lewis, and Thornhill, “Research methods for business students”, 5th edition, Pearson Education Limited, 2009, England.

[30] M. Hausi, J. Jens, S. Dennis, S. Margaret, T. Scott, and W. Kenny, “Reverse Engineering: A Roadmap”, ACM Special Interest Group on Software Engineering, Proceedings of the Conference on The Future of Software Engineering, pp. 47-60, 2000.

[31] M. Brian, S. Mancoridis and M. Traverso, “Searched Based Reverse Engineering”, ACM International Conference Proceeding Series, Proceedings of the 14th international conference on Software engineering and knowledge engineering, Vol. 27, pp. 431-438, 2002.

[32] Arie van, Deursen, and LizBurd, “Software Reverse Engineering”, Proceeding of 9th Working Conference on Reverse Engineering (WCRE), IEEE Computer Society Press, pp.1-4, 2002.

[33] E. Buss and J. Henshaw, “A software reverse engineering experience”, Proceeding of CASCON First Decade High Impact Papers, pp. 42-60, 2010

AUTHORS PROFILE

Rawan AbuLail received her B.Sc. degree from Philadelphia University, Amman, Jordan in 2002, and M.Sc. degree from Arab Academy for Banking and Financial Sciences in 2004, both in computer Her Ph.D. degree in computer information systems from Arab Academy for Banking and Financial Sciences, Amman, Jordan in 2009. Her research interests include Software Engineering, Agent Oriented Software Engineering, Artificial Intelligence, Decision Support Systems, and Information Security.

Mohammad Shkoukani received his B.Sc. degree from Applied Science University, Amman, Jordan in 2002, and M.Sc. degree from Arab Academy for Banking and Financial Sciences in 2004, both in computer His Ph.D. degree in computer information systems from Arab Academy for Banking and Financial Sciences, Amman, Jordan in 2009. His research interests include Agent Oriented Software Engineering, Software Reverse Engineering, System Analysis and Design, and Electronic Commerce Applications.