Analyzing Competences in Software Testing: Combining Thematic Analysis with Natural Language Processing (NLP)

Tajmilur Rahman, Joshua Nwokeji, Richard Matovu, Stephen Frezza, Harika Sugnanam, Aparna Pisolkar
Computer and Information Science
Gannon University, Erie, PA

{rahman007, nwokeji001, matovu001, sugnanam001, pisolkar001}@gannon.edu
sfrezza@franciscan.edu

***Abstract*—This Full Paper (Research) presents an analysis on the competences in software testing for the fresh graduates in computer science. Software Testing education (ST) is receiving increasing attention in literature, recent studies have evaluated instructional methods used in ST education. However, analysis of competences (skills, knowledge, and ability) required in ST education are lacking in literature. Competences play critical roles in curriculum development e.g., they inform the design of student learning outcomes, learning objectives and program outcomes. This full paper in the research category aims to analyze competences in ST education and then examine the gap between these competences and the current ST curriculum. Using natural language processing (NLP) techniques, we collect 2033 job descriptions from three popular job portals (indeed, monster, and career builder) in the USA and Canada. Also, we collected course syllabi from 20 universities offering ST courses and use these to assess the current curriculum in ST. We analyzed the data using thematic analysis and found that the current software testing curricula do not always teach or equip students with some of the soft skills they require to be successful in software testing career. For instance, our result shows that soft skills such as teamwork, communication, leadership, which are often required by software testing employers are not always taught in ST courses.**

***Keywords*—Software Testing, Competency, Testing Skills, Curriculum Development**

I. Introduction

One of the important goals in software development is to build a quality software that is reliable and devoid of defects. Software testing, a software quality assurance methodology, is used to achieve this goal. At each phase of the software development life cycle (SDLC), software testing plays vital roles, especially in the identification of defects, problems and points of failures in the software being developed [7] [24]. There has been a growing demand for software testing skills in the industry [6]. To meet these demands, academic institutions are beginning to integrate software testing in the computing curricular [6]. Software testing is currently taught as a core course in the software engineering curriculum by most institutions while some universities and colleges teach software testing as part of computer science and information systems.

Software testing education is also receiving attention in scholarship of teaching, some researchers are studying the various instructional methods used for teaching software testing in higher education. For instance, Krutz et. al., [10] studied project-based learning (PBL) as an instructional approach in software testing and reported that PBL enhanced students' learning. Similarly, in Deng et al [11], free and open source software (FOSS) was used to provide real-world testing experience to students. Other studies such as [6] focus on systematic reviews on software testing education while Chan et al [12] and Aniche et al [13] conducted surveys to understand students' perception of learning software testing.

Although existing studies make some meaningful contributions, they often neglect the competences needed in the ST profession. Competences support educational activities in various ways and, in particular, contribute to curriculum development. In ST education, competences can be used to develop student learning outcomes (SLO) which are core aspect of curriculum. ST competences can also inform the design of learning objectives found in syllabi as well as program outcomes found in academic majors [2].

Despite the critical roles they play in educational activities, competences are not well studied and are often neglected in software testing education. We found very few studies that have identified and analyzed the professional competences needed by employers of software testing. Similarly, studies that have examined the gap between software testing competences and the current course offerings in software testing are very few, if available at all. These create gaps in ST education research.

This study aims to identify, analyze and articulate the professional competences in software testing. Additionally, we identify the gaps between these competences and the current course offering in software testing. We believe that insight into software testing competences can strengthen teaching and learning of software testing. Hence the scope of our study is limited to these two research questions:

1) **RQ1:** What competencies (skills, knowledge, abilities) are required for a software testing professional?

2) **RQ2:** What is the gap between the required competences and the university courses that are currently offered in the institutions in the United States (USA) and Canada.

In order to answer these questions, we collect two sets of qualitative data. The first data set is job advertisements from the 3 job portals operating in USA and Canada. (see Table I). Our second data set is the course descriptions from the 20 leading institutions where software testing is taught. Data analysis is done using a combination of Natural Language Processing (NLP) and a Thematic Analysis. We anticipate that the competences identified from this research will play important roles in educational activities including curriculum development, course design, program development, etc. Also it will help ST educators understand the current gaps between the skills they currently teach in the classroom and what employers need so that they can re-design the syllabi accordingly.

The rest of this paper is structured as follows. In Section II, we describe the current literature on software testing education and provide an overview of competences. Section III describes our research methodology, including data collection, cleaning, processing, and data analysis. Results of our data analysis will be presented and discussed in Section IV. The limitations of our work will be discussed in SectionV. We finally conclude this paper and provide an overview of our future work in Section VI.

II. Literature Review

A. Overview of Competency

The term competence has been defined by many scholars in slightly different ways. However, most researchers agree that competences include three main components namely skill, knowledge and disposition [2], [16], [17]. Other synonyms are used by some authors to replace disposition, for instance, attitude was used in [16] while ability is used in [17]. For this research, we use ability instead of disposition. The definition of competency we used in this research is consistent with the definitions given in the existing studies [2], [16], [17].

Hence, we define competency as the skills, knowledge and abilities required to perform duties in each profession. Consistent with the classification of skills in Nwokeji et. al. [2], our definition further classifies skills into hard skills i.e., hands-on activities or practical tasks performed by an individual and soft skills i.e., personal attributes or qualities of an individual. Examples of hard skills include programming in a particular language e.g., Java, analyzing requirements, modeling processes, etc.; while examples of soft skills include communication, teamwork, and leadership abilities.

B. Related Works

Available research in software testing education can be categorized into three broad areas namely systematic literature mapping and surveys, instructional approaches, and curriculum development integration. The contributions of most studies (e.g., [6], [15]) that fall within the first category are limited to synthesis of existing literature in software testing education and mostly surveys (e.g., [12] [13]) to understand the issues and challenges in software testing education. For instance, the authors in [6] performed a systematic literature mapping on 204 publications in software testing education to highlight challenges and prospects. With regards to challenges, they found that courses within the software testing curriculum are not always aligned with industry needs.

In the second category, the authors examine the various instructional methods used in teaching and learning software testing. In this category, project-based learning [10], [11] appears to be popular. Others such as game-based learning approaches [19], [20] are also used. Project-based learning is an active learning approach that allows students to engage in projects to solve real-world problems and thereby acquire knowledge and skills [21], [22]. For instance, the authors in [10], [11] used open-source software in their courses to provide real-world software testing hand-on project experiences for their students.

We also found authors that carried out curriculum reform to align the course delivery to the industry needs [26]. Some of activities performed in this curriculum reform include improvement of the courses structure, teaching content and learning resources [26]. Some other studies in this category such as [27], [28] propose various approaches that can be used to integrate software testing into the computing curriculum.

While these studies make important contributions to the body of knowledge in software testing, they fail to analyze the competences (skills, knowledge, and dispositions) required in ST profession, and whether the current course offering/curriculum in ST deliver these competences. Hence, this study analyzes the competences in ST and examines the extend to which the current curriculum delivers these competences.

TABLE I
Job postings from job portals as of April 5, 2021

Job Site	Software Tester	QA Tester	Software QA
Indeed	153 - 117	121 - 54	143 - 54
Indeed CA	167 - 126	98 - 98	148 - 98
Monster	706 - 173	87 - 38	1099 - 279
Monster CA	158 - 59	706 - 162	165 - 61
Careerbuilder	496 - 196	759 - 344	182 - 138
Careerbuilder CA	20 - 12	150 - 12	12 - 12

III. Methodology

We followed a qualitative approach to conduct our study. We collected job posts from various job portals and academic course descriptions from various undergraduate institutions in US and Canada. We then analyzed the data to extract themes by performing a thematic analysis. Thematic analysis is one of the most popular analysis approaches on qualitative data useful for identifying, analyzing and interpreting patterns of the integrated meaning within the qualitative data [14].

Our collected data is textual and we needed to analyze the texts and key-phrases to interpret them into meaningful

TABLE II
SOFTWARE TESTING RELATED COURSES USED IN OUR STUDY

University	Course Code - Title
University of Virginia	CS 3250 - Software Testing
Rochester Institute of Technology	SWEN 352 - Software Testing and Analysis
University of British Columbia	CPEN 422 - Software Testing and Analysis
University of Florida	CEN 4072 - Software Testing & Verification
George Mason University	SWE 637 - Software Testing
Harvard University	CSCI E-19 - Software Testing and Test-Driven Development
University of Pennsylvania	CIS 700 - Software Analysis and Testing
Depaul University	SE 333/433 - Software Testing and Quality Assurance
Georgia institute of technology	CS 6340 - Advanced Topics in Software Analysis and Testing
University of Calgary	SENG 637 - Dependability, Reliability, and Testing of Software Systems
Rensselaer Polytechnic Institute	CSCI.4962 - Software Verification
Clemson University	CpSc 873 - Software Verification, Validation and Measurement
University of Houston–Clear Lake	SWEN 5431 - Testing, Verification & Validation Syllabus
University of California San Diego	CSE-41266 - Software testing for quality Assurance
Naval Postgraduate School	SW 4540 - Software Testing
Arizona State University	CSE 464 - Software Quality Assurance and Testing
Montgomery College	ITI 232 - Software testing and quality assurance QA level-1
De Anza College	CIS 74 - Software Quality Assurance
Txstate	CS 4379A : Software Testing
Bellevue college	ISIT 324 - Software Testing

keywords indicating skills, abilities, and knowledge required for software testing jobs in the observed job market.

A. Data Collection

We collected job posts from 3 major popular job sites in the USA and Canada i.e., "Indeed", "Monster", and "Careerbuilder". These three portals have USA and Canada sites. In total, we collected data from six job sites (USA and Canada for each three). Table I shows how many job postings we could collect from each of the job sites. We used a custom Python script to crawl the posted job advertisements. The search terms used in the script were "Software Tester", "QA Tester", and "Software QA". These search terms were determined by first manually observing the titles of the job posts.

To collect the university course descriptions, we considered a wide range of diversity in terms of institution type and Computer Science/IT/Engineering ranking. We selected 14 universities, 2 polytechnic institutes, and 4 colleges where software testing is taught in the undergraduate programs. 12 of the selected institutions had a ranking below 100, 2 were ranked between 100 - 200, another 2 were ranked between 200 - 400 and the last 4 had no ranking.

We first searched for the top institutions ranked from 1 to 400 according to the website "usnews.com" [29]. We then browse their undergraduate programs in software engineering and/or computer science discipline. This process was conducted manually using "Software Testing", "Software Quality", and "Quality Assurance" as the search terms. Courses containing one or more of these terms in the course title were selected. We also manually verified them by reading the course description. Table II shows the universities and course details that were considered after the verification process.

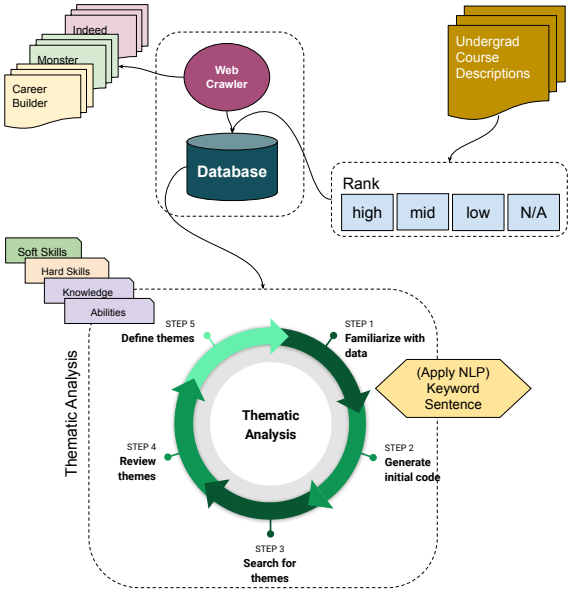

Fig. 1. Methodology at a glance: Data collection and thematic analysis

B. Data Cleanup and Pre-processing

Given all the job posts collected using our search keyphrases, shown in Table I, we first selected postings that were relevant to software testing and quality assurance. We then filtered out posts that weren't entry-level to avoid having job experience as an x-factor in our study. This eliminated posts for experienced testers that may require skills or knowledge that aren't obviously taught in an undergraduate-level program.

We also removed the duplicate postings, based on "job title", "company name", "location" and "expected start date", since many companies post the same job in multiple job sites. We then applied a systematic random process to avoid bias, as was done by Nwokeji et al. [2]. The selected job postings were then stored into a database for further analysis and keyword extraction.

Since the course descriptions were collected manually, we did not spend much effort on cleanup. However, we still went through multiple rounds of eye-balling over the description text to ensure that all courses were focused on software testing and/or software quality assurance.

C. Data Analysis

Our next step was to find out the presence of keywords related to software testing. For this, we looked into the "job description", "required skills", and "job responsibilities" to

TABLE III
AN EXAMPLE OF A HARD-SKILL AND A SOFT-SKILL EXTRACTED FROM THE JOB POSTINGS

Skill	Keyword	Count	Sentences (first 3)
Hard	SQL	32	"Sound knowledge of database and SQL in an oracle environment is a must.", "Experience testing SQL Server/Oracle databases and strong knowledge of SQL queries to test data validity...", "various knowledge and skills in scripts, SQL, Operating System, Networking, etc"
Soft	Reliability	8	"you will test web software for the validity of results, accuracy, reliability and compliance with established standards.", "engineers to build quality into our product from design to launch Maintain the integrity and reliability of Go1's automation test.", "testing to ensure system reliability, capacity, and scalability"

identify the required skills, abilities, and knowledge. We identified the relevant themes i.e., skills, abilities, and knowledge, from the job description text (general description, required skills, job responsibilities). Figure 1 summarizes our process of data collection and thematic analysis.

The thematic steps [1], [3] include i) familiarizing with data ii) generating initial codes, iii) searching for themes iv) reviewing themes, and finally, v) defining the themes. We familiarized ourselves with the data during the data cleanup step described in the previous section. After familiarization, we performed our thematic analysis and applied NLP to identify job description text that contain our desired features. This is similar to the initial code extraction as defined by Bruan and Clarke [1]. The features in our data are keywords that classify competence into skills, knowledge and ability, with skills further classified into "Soft-Skills" and "Hard-Skills".

Once the features were classified, we searched for themes to find relevant information useful to answer our research questions. The key themes in our research questions are "competency", "skill", "knowledge", "ability", "required", "software testing", "university courses", and "professional". Braun et al. [1] used inductive and deductive approaches to search for themes. The choice between the two approaches depends on whether we are interested to use the selected themes to answer predefined research questions or we want the selected themes to formulate research questions or theories. We followed the deductive approach since our study is more related to the latter.

Key Features and Themes:

Theme represents the core concept of the data, i.e., what was the core meaning when a keyword was used in a sentence. The theoretical framework of our study forms a relationship among the key features in our research questions. We are interested in finding out what features are required from the job posts, and what features are available in the university courses. These three top level features i.e., skills, knowledge, and ability, correspond to our core theoretical framework.

Skills represent the expertise on a specific tool or practice. Hence, skills are proper nouns. Skills related to hands-on expertise are considered to be "Hard Skills" while skills that are human attributes and can be improved or achieved by practice, are "Soft Skills" e.g., communication skill. We want to see the skills mostly required by the employers when they are looking for a fresh graduate as a software tester. Furthermore, we want to see which type of skills are more frequently looked for by the employers. From our NLP analysis output we collect the proper nouns that appeared with "Experience" or "Expert" or "Expertise" to bucket them as skills.

Knowledge represents what a candidate needs to know about. Employers may look for if the candidate has proper understanding on a topic, or has a clear idea on the subject matter which can be a process, technique, strategy, method etc. Asking for knowledge can be very close to asking for skill. For example, knowledge on continuous integration maybe looking for whether the candidate has expertise on setting up continuous integration system. It may also indicate whether the candidate has an understanding on what is continuous integration and why it is required etc. To collect the Knowledge keywords, we consider the proper nouns associating with the feature "Knowledge".

Ability indicates whether a person is able to do something in a certain way e.g., ability to work independently. Here, the keyword is "Independent". In most of the cases ability refers to the adverb form of the keyword. Therefore, we extract the adverbs from our NLP analysis to get the ability keywords.

Once we run our NLP engine on the job posts for three features "Skills", "Knowledge", and "Ability", then we obtain the proper noun keywords associated with "Skills", proper noun keywords associated with "Knowledge", and adverb keywords associated with "Ability". NLP engine selects only those sentences where the respective keyword (pronouns, and adverbs) is present for the three core themes.

Table III shows an example of a hard-skill and a soft-skill extracted using our NLP engine. The last column in Table III shows the first three sentences from the database to illustrate how the keywords are used in the job postings.

Review Themes:

In the initial list of keywords, hard and soft skills both appeared as mixed. We reviewed all the keywords and classified them into Hard-Skills, and Soft-Skills. Please note that some keywords could appear under multiple features. For example, the keyword SQL in Table III appeared with "Knowledge" in the first sentence, and with "Experience" in the second sentence. Therefore, SQL appeared in "Knowledge" as well as "Skill". For such keywords, we appropriately classified them e.g., SQL was moved from "Knowledge" to "Hard-Skills".

We continued our review for the other theme "Ability" where adverbs were identified by our NLP engine. Most of the abilities appeared as "*ity" in the sentences. We converted them into the base words. For example, "*ability to effectively write test cases*", effectiveness is an ability and we stored the keyword as "Effective".

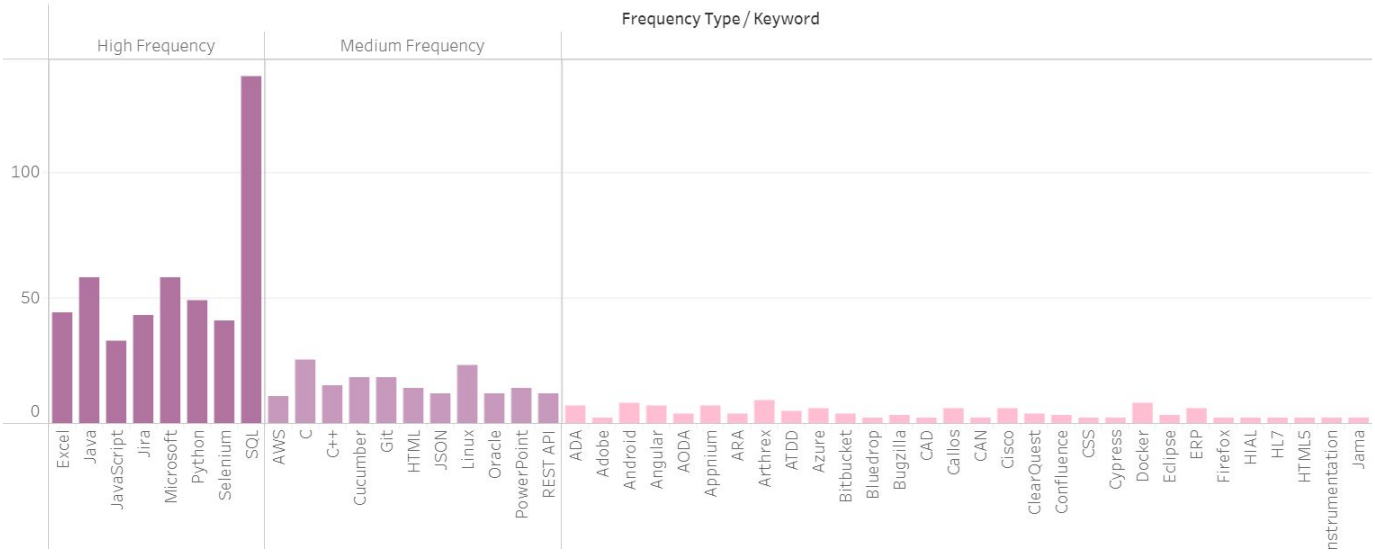

Fig. 2. Hard-skills required in job posts grouped into three frequency-groups

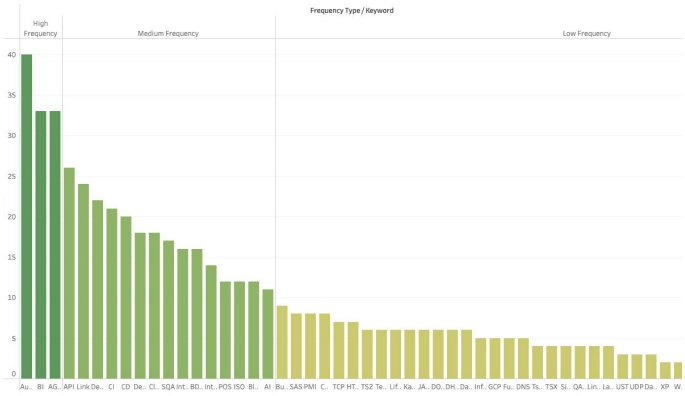

Fig. 3. Knowledge keywords required in job posts grouped into three frequency-groups

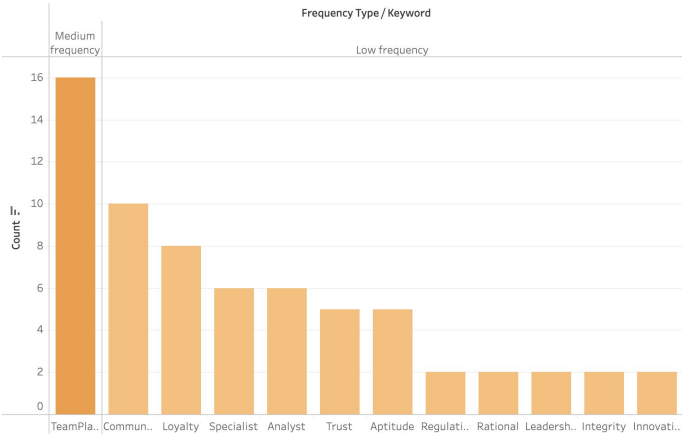

Fig. 4. Soft-skills required in job posts grouped into three frequency-groups

Finally, we grouped the keywords in each bucket into High, Medium, and Low frequency groups based on how many times they appeared in the job posts data set. We did not consider the keywords appearing only once. We also ignored the obvious comparison keywords (e.g., “test”, “testing”, “QA”, “quality assurance”, etc.) because they must be present in both university course description and job posts.

D. Compare Keywords with University Courses

We wanted to compare how many keywords from the four buckets identified above are found in the keywords extracted from the course descriptions. For this, we observe how many of the required skills were found in the course descriptions.

We then apply NLP to extract all the keywords from the university course descriptions, similar to what we did for job posts. We got returned keywords such as “software development”, “reliability”, “Black Box”, “Unit Test” etc. We removed the obvious keywords such as “software testing”, “software quality”, “testing process” etc. After the removal process, we ended up with 510 keywords containing sentences.

E. Credibility of Thematic Analysis

Reliability is an important aspect of research, as it ensures trust, confidence and trustworthiness of research results or findings [3]. In quantitative studies, reliability may be achieved using statistical methods. However, since qualitative studies are not statistically oriented, credibility has been proposed as an alternative to reliability for qualitative studies [3]. Peer debriefing is one technique used to ensure credibility in qualitative studies, see [3], [4].

Hence we used peer debriefing for this study. For the debriefing, we first identified an experienced colleague (a professor and researcher from the same institution with the authors) who did not take part in thematic analysis of this paper. Then we set up a meeting to debrief our coding process with him. Afterwards, we shared the transcript of the coding

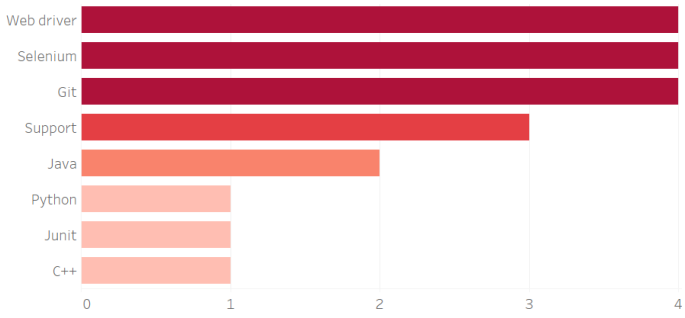

Fig. 5. Hard-skills mapped in university course descriptions

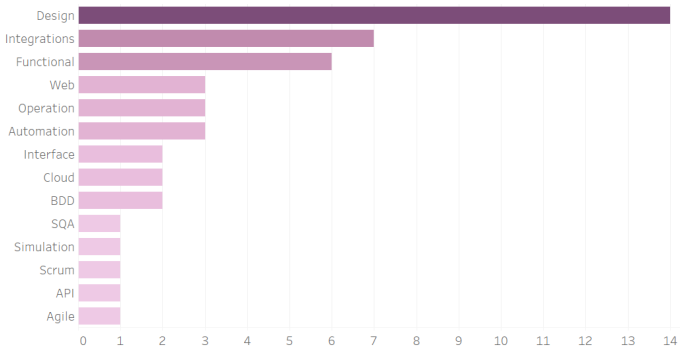

Fig. 6. Knowledge mapped in university course descriptions

with him to review, identify any inconsistency or error and provide feedback.

IV. Results and Discussion

In this section, we present the results and findings from our data analysis and use these to answers the two research questions presented in Section I.

A. *RQ1: What are the competences required in the software testing profession*

In our literature review section, we defined competence as skills, knowledge and abilities. Figure 2 shows the hard-skills grouped into three frequency groups i.e., high, medium, and low based on the thresholds of 10 and 30. Any keywords appearing with frequency less than 10 belongs to low and above 30 belongs to high frequency group. Due to space limit keywords with frequency less than 5 are trimmed from the figure. In Figure 2, we see that SQL is the most demanded hard skill for software tester positions. There are few other hard-skills in this group such as, Java, Javascript, Microsoft tools especially Excel, Selenium, Python, and Jira (project management tool).

Figure 4 shows the soft-skills required by the job posts. The only soft-skill "Team Player" belongs to the medium frequency group. There is no keyword in the high-frequency group. Software testing primarily involves teams which requires software companies look for someone who has such team playing skill.

Similar to the hard-skills, and soft-skills, "knowledge" keywords are also grouped into three groups as we can see in Figure 3 where Automation (Au.), Business Integration (BI), and Agile (AG) are the most frequently demanded keywords in terms of knowledge. Similarly "team playing" is one of the highly demanded soft-skills, and ability to test effectively, and independently are the "abilities" mostly demanded by the employers found in software testing job posts.

B. *RQ2: What is the gap between the required competences and the university courses that are currently offered in the institutions in the USA and Canada*

Our objective is to observe how many of these keywords we also find in the course descriptions as the learning objectives or prerequisites for that course. We consider prerequisites besides learning objectives because we understand that programming language is usually not a learning objective in a testing course even though Java/Python are the most demanded hard-skills in the industries. Most of the cases, languages are taught separately in another course as part of the same program.

Figure 5 shows the mapping between the required hard-skills found in the job posts and the university course descriptions. Here, we see that "Selenium" is one of the most highly demanded hard-skills which strongly appears to be taught in the universities as well. On the other hand, another hard-skill "Web driver" has strongly appeared in the university course descriptions, which is one of the least frequent keywords in the job posts (doesn't appear in Figure 2 since its frequency is below 5). Other hard-skills found in the university courses are "Git", "Support" (tool support), "Java", "Python", "JUnit", and "C++" in the order of strength of mapping. We do not see a wide verity of hard-skills in the course descriptions that cover the required hard-skills in the job posts. There are many hard-skills from high and medium frequency groups are not present in the university course descriptions, such as, "JavaScript", "Jira" (project management tool), "Microsoft" (Microsoft office and testing tools), and SQL from the high-frequency group, and "AWS", "C", "Cucumber" (testing tool), "HTML", "JSON", "Oracle", and "REST API" from the medium frequency group. There are still many keywords did not appear in the university course descriptions from the low frequency group of hard-skills required in the job posts.

Similarly, there are high frequency "knowledge" keywords that strongly appear in course descriptions such as, "Design" (knowledge on designing test cases), "Integration" (knowledge on test integration), and "Functional" (knowledge on functional testing) as shown in Figure 6. Knowledge on test case design is found in the university course descriptions exceptionally high number of times. Which is understandable since this is the most basic topic to learn in software testing. Soft-skills from the job posts do not appear at all in the course descriptions. This is a large gap we observed between the requirements in industries and the university course curricula.

Figure 8 shows the mapping of the "ability" keywords between job posts and the course descriptions. In terms of job posts, abilities reflect how much capable the candidate is to perform a task or how well a test can be performed. A very

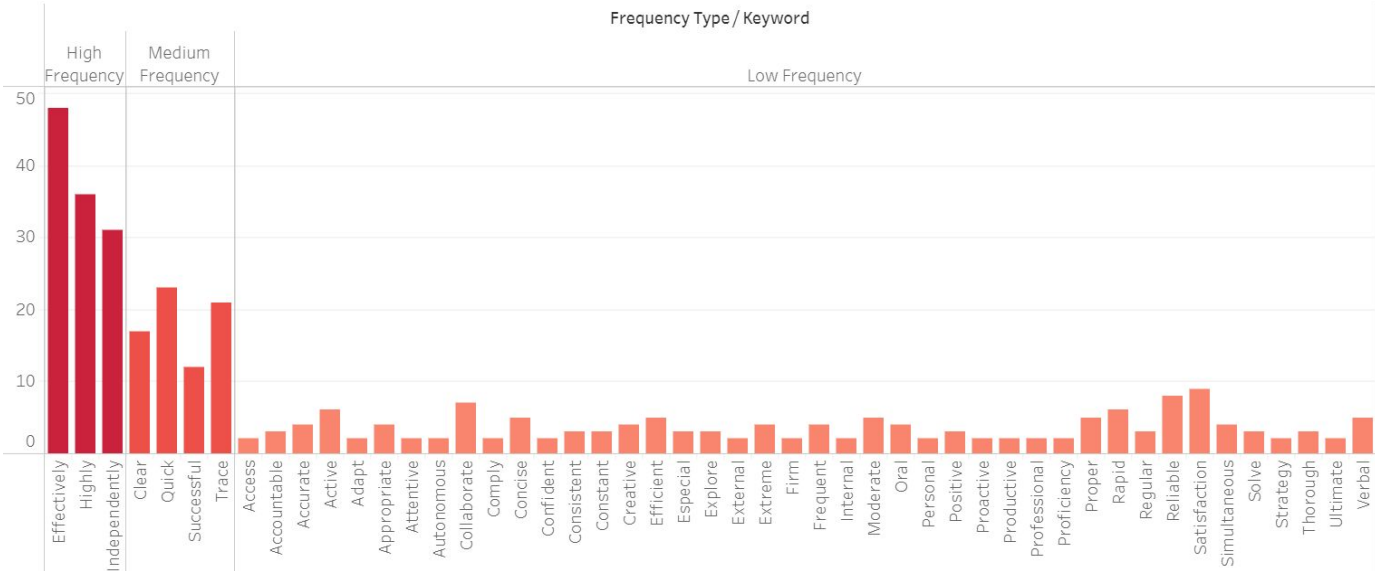

Fig. 7. Abilities required in job posts grouped into three frequency-groups

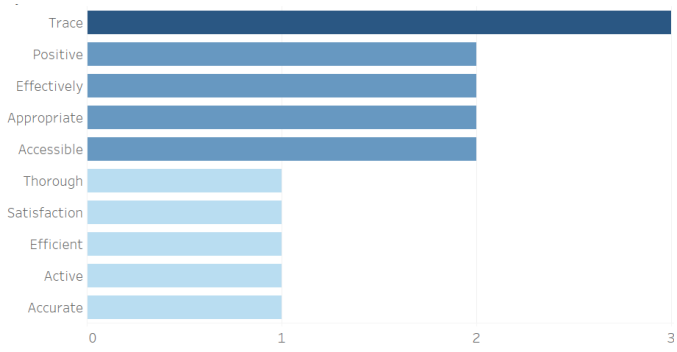

Fig. 8. Abilities mapped in university course descriptions

common ability in testing is "traceability". Traceability indicates that the test case is traceable, meaning, it is transparent and quite clear which feature/use-case a particular test-case written for. "Traceability" is one of the medium-frequency keywords in the job posts which is also strongly found in the course descriptions.

Figure 5, 6, 8 indicate a clear gap between the job posts and the university course descriptions. Only a few of the requirements keywords from the job posts mapping with the university data. A very low percentage of 8.6% of the hard-skills, 20% of the abilities, and 25% of the knowledge keywords did match with the university course descriptions.

Many cutting edge technologies facilitate testing in industries are not currently taught as part of software testing curriculum. For example, SOAP and REST APIs, simulation techniques, testing frameworks are not mentioned in the university course descriptions. Although, API in general and Simulation are found to be taught only in one university. Yet, simulation techniques are very much necessary for testing complex systems and we expect to see this being taught in many universities at the undergraduate level.

There are 17 soft-skills identified in the job posts and unfortunately we did not find any of these keywords mentioned in the university course descriptions. We therefore, see here is a large gap where our students are not familiar with such soft skills and they do not get an opportunity to develop them especially to apply in a testing team.

Similarly, there are 50 abilities identified as required by the employers. Some of the high in frequencies are, "Effectively", "Highly", and "Independently", found in many of the job posts to invite candidates who have the ability to work effectively, independently, and efficiently. The medium frequency keyword "trace" (traceability) has been found in three course descriptions while effectiveness and efficiency found in two and one course descriptions respectively. We believe that such abilities need to be practiced among the students at the undergraduate level and the testing related courses must introduce or reinforce such abilities in the respective course-projects or assignments.

V. Limitations & Threats to Validity

In this study, we use course syllabi from 20 institutions to assess the current curriculum and course offerings in ST. We recognize that syllabus, especially collected from a limited number of institutions, may not adequately describe ST curriculum or courses. Possibly, other data sources such as lecture notes and assessment may provide a detailed description of the curriculum and course offerings in ST. Yet, the scope of our study is beyond these data sources.

Similarly, job advertisement may not fully represent the competences employers desire in ST professionals. Conducting interviews with the employers may provide a more detailed set competences. In the next section, we explain how we plan to address these limitations in our future work.

VI. Conclusion & Future Work

Although teaching and learning software testing is becoming popular in universities, assessment of competences required in software testing professions is still lacking in literature. Also, it is not clear whether the current curriculum aligns with the competences required by employers. Periodic assessment of competences is very vital to quality curriculum and courses delivery as well as student career success.

Our aim in this research is to identify the current gap in software testing literature, accordingly we combine natural language processing with thematic analysis to analyze 2033 job posts to identify the competences (skills, knowledge, and abilities) required in the software testing profession. Furthermore, we examine if the current courses in software testing are actually delivering the competences we identified from job posts. To do this we analyzed the course descriptions from the 20 diverse educational institutes in the USA and Canada including Harvard University and University of Pennsylvania.

Based on our data analysis, we conclude that universities are beginning to integrate software testing into the computing curricular. However, the current software testing curricular do not teach or equip students with the soft-skills they require to be successful in software testing career. For instance, our result show that soft-skills such as teamwork, communication, leadership, which are required by software testing employers are absent in the course syllabi we studied. Furthermore, many hard-skills, knowledge and abilities required in the software testing profession have not been properly integrated in the curricular. For instance, very few universities are teaching hard-skills (e.g., Jira, Cloud Computing, JavaScript, Linux) as part of software testing curriculum. The same applies to knowledge (e.g., continuous integration, and continuous delivery etc.) and abilities (e.g., "Independent work", "Reliability"). This indicates that the software testing courses in the undergraduate programs mostly focus on the hard-skills and knowledge, and a very little on abilities. Yet, many cutting-edge tools and techniques are not being introduced/reinforced in the undergraduate level to make our students prepared before they get into the professional fields.

Our future work will focus on addressing the limitations we described in the previous section. We plan to collect other educational documents e.g., lecture materials, assessments, program outcomes etc.; and also interview selected ST instructors. This would ensure a more detailed assessment of ST curriculum and course offerings. We also plan to increase the number of institutions wherein we collected data. Furthermore, we plan to validate the competences derived from job advertisement by interviewing employers and recruiters.

References

[1] Braun, Virginia, and Victoria Clarke. "Using thematic analysis in psychology." Qualitative research in psychology 3, no. 2 (2006): 77-101.

[2] Nwokeji, Joshua Chibuike, Richard Stachel, Terry Holmes, and Rita O. Orji. "Competencies Required for Developing Computer and Information Systems Curriculum." In 2019 IEEE Frontiers in Education Conference (FIE), pp. 1-9. IEEE, 2019.

[3] Nowell, Lorelli S., Jill M. Norris, Deborah E. White, and Nancy J. Moules. "Thematic analysis: Striving to meet the trustworthiness criteria." International journal of qualitative methods 16, no. 1 (2017): 1609406917733847.

[4] Belotto, M. J. "Data analysis methods for qualitative research: Managing the challenges of coding, interrater reliability, and thematic analysis". Qualitative Report, 23(11) (2018).

[5] Malterud, K. (2001). Qualitative research: standards, challenges, and guidelines. The lancet, 358(9280), 483-488.

[6] V. Garousi, A. Rainer, P. Lauvås Jr, and A. Arcuri, "Software-testing education: A systematic literature mapping," J. Syst. Softw., vol. 165, p. 110570, 2020.

[7] M. Tuteja, G. Dubey, and others, "A research study on importance of testing and quality assurance in software development life cycle (SDLC) models," Int. J. Soft Comput. Eng., vol. 2, no. 3, pp. 251–257, 2012.

[8] S. Heggen and C. Myers, "Hiring millennial students as software engineers: a study in developing self-confidence and marketable skills," in Proceedings of the 2nd International Workshop on Software Engineering Education for Millennials, 2018, pp. 32–39.

[9] F. Gurcan and N. E. Cagiltay, "Big data software engineering: Analysis of knowledge domains and skill sets using LDA-based topic modeling," IEEE Access, vol. 7, pp. 82541–82552, 2019.

[10] D. E. Krutz, S. A. Malachowsky, and T. Reichlmayr, "Using a real world project in a software testing course," in Proceedings of the 45th ACM technical symposium on Computer science education, 2014, pp. 49–54.

[11] L. Deng, J. Dehlinger, and S. Chakraborty, "Teaching Software Testing with Free and Open Source Software," in 2020 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW), 2020, pp. 412–418.

[12] F. T. Chan, T. H. Tse, W. H. Tang, and T. Y. Chen, "Software testing education and training in Hong Kong," in Fifth International Conference on Quality Software (QSIC'05), 2005, pp. 313–316.

[13] M. Aniche, F. Hermans, and A. Van Deursen, "Pragmatic software testing education," in Proceedings of the 50th ACM Technical Symposium on Computer Science Education, 2019, pp. 414–420.

[14] Castleberry A, Nolen A. "Thematic analysis of qualitative research data: Is it as easy as it sounds?," Currents in Pharmacy Teaching and Learning. 2018 Jun 1;10(6):807-15.

[15] L. P. Scatalon, J. C. Carver, R. E. Garcia, and E. F. Barbosa, "Software testing in introductory programming courses: A systematic mapping study," in Proceedings of the 50th ACM Technical Symposium on Computer Science Education, 2019, pp. 421–427.

[16] M. Daniels et. al., "Competencies for paradigm shift 'survival'," in 2015 IEEE Frontiers in Education Conference (FIE), 2015, pp. 1–6.

[17] E. Moustroufas, I. Stamelos, and L. Angelis, "Competency profiling for software engineers: literature review and a new model," in Proceedings of the 19th panhellenic conference on informatics, 2015, pp. 235–240.

[18] D. E. Krutz, S. A. Malachowsky, and T. Reichlmayr, "Using a real world project in a software testing course," in Proceedings of the 45th ACM technical symposium on Computer science education, 2014, pp. 49–54.

[19] B. S. Clegg, J. M. Rojas, and G. Fraser, "Teaching software testing concepts using a mutation testing game," in 2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering Education and Training Track (ICSE-SEET), 2017, pp. 33–36.

[20] P. H. D. Valle, A. M. Toda, E. F. Barbosa, and J. C. Maldonado, "Educational games: A contribution to software testing education," in 2017 IEEE Frontiers in education Conference (FIE), 2017, pp. 1–8.

[21] J. C. Nwokeji and S. T. Frezza, "Cross-course project-based learning in requirements engineering: An eight-year retrospective," in Proceedings - Frontiers in Education Conference, FIE, 2017, vol. 2017-Octob.

[22] J. C. Nwokeji, F. Aqlan, A. Olagunju, T. Holmes, and N. C. Okolie, "WIP: Implementing Project Based Learning: Some Challenges from a Requirements Engineering Perspective," in 2018 IEEE Frontiers in Education Conference (FIE), 2018, pp. 1–5.

[23] J. C. Nwokeji, R. Matovu, and B. Rawal, "The use of Gamification to Teach Cybersecurity Awareness in information systems," 2020.

[24] O. A. L. Lemos, F. F. Silveira, F. C. Ferrari, and A. Garcia, "The impact of Software Testing education on code reliability: An empirical assessment," in Journal of Systems and Software, 2018, pp. 497–511.

[25] L. P. Scatalon, M. L. Fioravanti, J. M. Prates, R. E. Garcia, and E. F. Barbosa, "A survey on graduates' curriculum-based knowledge gaps in software testing," in 2018 IEEE Frontiers in Education Conference (FIE), 2018, pp. 1–8.

[26] B. Zhu and S. Zhang, "Curriculum reform and practice of software testing," in 2013 the International Conference on Education Technology and Information System (ICETIS 2013), 2013, pp. 846–849.
[27] T. B. Hilburn and M. Townhidnejad, "Software quality: a curriculum postscript?," in Proceedings of the thirty-first SIGCSE technical symposium on Computer science education, 2000, pp. 167–171.
[28] E. L. Jones, "An experiential approach to incorporating software testing into the computer science curriculum," in 31st Annual Frontiers in Education Conference. Impact on Engineering and Science Education. Conference Proceedings (Cat. No. 01CH37193), 2001.
[29] US News. [Online] Available: "https://www.usnews.com/education/best-global-universities/canada". Accessed: April 2021.