Investigating the Skill Gap between Graduating Students and Industry Expectations

Alex Radermacher, Gursimran Walia, and Dean Knudson
North Dakota State University
IACC Bldg., Room 258 A20
Fargo, North Dakota, United States of America
{alex.radermacher, gursimran.walia, dean.knudson}@ndsu.edu

ABSTRACT

Graduating computer science and software engineering students do not always possess the necessary skills, abilities, or knowledge when beginning their careers in the software industry. The lack of these skills and abilities can limit the productivity of newly hired, recent graduates, or even prevent them from gaining employment. This paper presents the results of an empirical study where twenty-three managers and hiring personnel from various software companies in the United States and Europe were interviewed. Participants were asked about areas where recent graduates frequently struggled when beginning employment at their companies and which skill deficiencies might prevent a recent graduate from being hired. The results of this study indicate that recent graduates struggle with using configuration management systems (and other software tools), effectively communicating with co-workers and customers, producing unit tests for their code, and other skills or abilities. The results also indicate that a lack of project experience and problem solving abilities are the most commonly cited issues preventing students from gaining employment. This research is intended to assist educators in identifying areas where students may not measure up the expectations of industry companies and in improving the curriculum at their universities to better prepare them for their future careers.

Categories and Subject Descriptors

K.3.2 [**Computers and Education**]: Computers and Information Science Education—*curriculum*

General Terms

Human Factors

Keywords

Software developer, Required skills, Computer science pedagogy, Computer science education

ICSE Companion'14, May 31 – June 7, 2014, Hyderabad, India
Copyright 2014 ACM 978-1-4503-2768-8/14/05...$15.00
http://dx.doi.org/10.1145/2591062.2591159

1. INTRODUCTION

When computer science and software engineering students graduate from their undergraduate programs, they are not always prepared for their future careers in the software industry [3]. This lack of preparation is not just limited to programming skills or other computer science concepts, but can also include communication ability, familiarity and proficiency with software tools, and an understanding of software engineering concepts such as software development processes. Recently graduated students who are beginning their first job may struggle in these or other areas [2, 4, 7, 14, 20]. In some cases, there is a large enough gap between students' skills and the expectations of industry managers or other hiring personnel, that it may prevent students from receiving jobs.

At North Dakota State University (NDSU), students are required to take a capstone project course during the final semester of their senior year. In this course, students work on "real-world" projects for various industry companies that develop software [12]. Over the past several years there has been an effort to incorporate more of the tools and practices used in industry into the course in order to expose students to concepts, tools, etc. that they will likely use in their future jobs after graduation. Some of examples of these include the use of configuration managements tools such as Subversion and the use of agile software development processes and techniques [13].

One of the companies that had worked with NDSU on a capstone project had raised concerns with the capstone instructor about some of the students from the university who had applied for positions at their company. The company mentioned that these students lacked the ability to use various tools and did not possess conceptual knowledge that was necessary for employment at their company. In order to better understand the extent to which these issues were prevalent at NDSU and other universities, a systematic literature review was conducted to identify common areas where recently graduated students struggled when beginning their first jobs. The review found a wide variety empirical evidence identifying ways in which recent graduates struggled in their new jobs [21]. To unite the different definitions and criteria used by other researchers in identifying these issues, the term *knowledge deficiency* was developed to encompass these previous definitions.

Although the literature review exposed several knowledge deficiencies, many of the studies included in the review did not contain enough descriptive information to suggest pos-

sible solutions for those knowledge deficiencies. In order to gain a better understanding of knowledge deficiencies, managers and hiring personnel at various software companies were interviewed regarding their perception of knowledge deficiencies that recent graduates had experienced when working at their company. Twenty-three respondents provided information about areas where their new hires struggled when beginning their job, as well knowledge deficiencies that prevented recent graduates from being hired by the company. The results of these interviews serve as the main focus of this research, but a comparison between these results and those from previous empirical research into this subject is also given. This comparison serves to strengthen the results from this research and to provide educators with an updated look at areas in which industry's needs are not being served. It also provides additional qualitative information about knowledge deficiencies, which was not present in many previous studies examining the gap between student skills and employer expectations.

The remainder of the paper is structured as follows: Section 2 presents related work from other researchers. Next, section 3 describes the study design used in this research. Following this, the results of the study are presented in section 4. Section 5 discusses the study results. Threats to the validity of the study are described in section 6. Finally, section 7 provides a conclusion and lists future research work.

2. BACKGROUND AND RELATED WORK

To gain a better understanding of the type of knowledge deficiencies that existed in graduating computer science students, a systematic literature review was conducted [21]. A systematic literature review, as opposed to an ad hoc review, is a formally defined process that establishes a series of research questions, evaluation criteria, and other predefined qualities to ensure that the papers included in the review are relevant and of high quality. The researchers of this study used the protocol established by Kitchenham when conducting the systematic literature review [10, 11]. This section reports the major results from that literature review and provides a description of knowledge deficiencies uncovered by other researchers. A complete description of the systematic literature review process, detailed descriptions of the knowledge deficiency categories, and additional information can be found in [21].

Several researchers have investigated various ways in which recently graduated students struggle when beginning their first jobs. One of the more common forms was through interviews and surveys of industry managers and other hiring personnel. In a 1996 study, Byrne, et al. conducted interviews of sixteen project, department, or personnel managers at various software companies in Ireland [4]. Roughly half of the respondents indicated that additional emphasis should be placed on software testing, communication skills, and understanding the software development life-cycle. In a study where twenty industry professionals were interviewed, Simmons, et al. also reported that students needed to be familiar with software development processes and that curricula should put more emphasis on requirement gathering and elicitation [25].

Miller, et al. conducted a study where ten IT professionals in charge of hiring were interviewed about necessary competencies for applicants [20]. The interviewed professionals were asked about the importance of four preselected competencies (e.g. abstraction and domain modeling) and also provided additional unsolicited responses about areas where applicants needed to be competent such as problem solving and teamwork experience. In a 2009 study, McGill investigated the types of skills needed in the game development industry to determine if there was a gap between industry expectations and academic programs for game development [18]. Twenty-six hiring personnel or other professionals in the game development industry and fifteen educators from universities with a game development program participated in a survey. The results from the study found that industry placed a greater emphasis on multi-threaded programming, tool development, and programming languages such as C++ and Lua than academia did.

Another popular method of identifying common issues and problems experienced by newly hired recent graduates involved the use of case studies. Begel, et al. conducted a case study of eight new software developers at Microsoft over the course of two months [2, 3]. The researchers of that study found that the new software developers most commonly experienced difficulties with communication and collaboration. One specific example was that several of the new developers were averse to asking questions even when they were struggling with an issue. Several technical issues were also reported as a common source of difficulty. The most frequent was the use of the revision control system, with which six of the eight subjects experienced some form of trouble. Four of the subjects were reported to have struggled with using debugging tools effectively when attempting to fix bugs in their code.

Haddad also used a case study to assess newly hired computer science students at a small software development company [7]. The study primarily focused on evaluating the cognitive abilities of fourteen developers who had been hired out of college within the past year, but also assessed some technical skills of the subjects. The results of the study indicated that subjects generally had issues with algorithm analysis and design, an inability to provide alternative solutions to problems, and a tendency to perform minimal and ineffective amounts of testing. Haddad also reported that the subjects had acceptable technical skills and a good understanding of fundamental computer science concepts and programming principles, but that the subjects did not always effectively apply these skills to code development or algorithm design.

Lethbridge examined knowledge deficiencies by surveying over one hundred software professionals about their current level of knowledge and proficiency in a large number of different topics and comparing those results against respondents' rankings of those areas when they had just graduated from college [14–16]. The results from these studies included several areas where professionals indicated that they needed to learn much more about a topic than they had been taught in college. These topics included data structures, software testing, project management, object-oriented concepts, configuration management tools, and software design among others.

Other researchers have also approached this issue by evaluating whether or not senior-level computer science and software engineering students are proficient at some activity commonly performed in the software industry. Eckerdal, et al. conducted a large study at multiple universities to determine if senior students were capable of producing good software designs [6]. The researchers of the study found that only 9% of students were capable of producing acceptable de-

signs. A follow-up study found similar results, but did indicate that students did have the ability to recognize good design [17]. Sudol, et al. examined misconceptions about software development held by students and found that students held misconceptions about using encapsulation and taking advantage of implementation details in code, and whether software could be successful if it shipped with defects [26]. Carver, et al. conducted a study to determine if senior-level students were capable of producing a complete test suite for a simple program both with and without the aid of code coverage tools [5]. The results from the study indicate that without the use of a code coverage tool students were not able to create a complete test suite. When a code coverage tool was used, students were able to create a complete test suite, but produced a large number of redundant test cases, in some cases upwards of three times as many as necessary.

Researchers in fields related to computer science and software engineering have also conducted studies that are relevant. McGuire, et al. measured the gap between how IS professionals rated their current competency in several areas of process improvement (such as those in the Capability Maturity Model) and their perceived importance of those areas [19]. One hundred fourteen IS professionals working on software development projects participated in the study. The results indicated that these professionals felt as though their knowledge in the areas of risk analysis, process control tools, change management, metrics and measurement, and other categories was substantially lower than the importance that was placed on these categories. Scott, et al. measured the gap between the skills of graduating IS students and the expectations of companies in Africa [23]. Their research found that databases were the area with largest gap between the abilities of graduating students and industry expectations. Trauth, et al. conducted a study to determine the difference placed on the importance of several categories (e.g. Programming Languages, Networking, System Development) by both academia and industry [28]. The results indicated that academia and industry were not always aligned, with academia putting too much focus in teaching a specific programming language and project implementation, and not enough in areas such as networking and systems integration. Although these studies are not specifically about computer science and software engineering students, they cover many of the same topics and help provide additional insight into areas where knowledge deficiencies may exist.

After extracting knowledge deficiencies identified in the existing literature, a taxonomy was created to classify deficiencies into multiple categories. The high-level categories include *Computer Science Concepts*, *Software Tools*, *Software Engineering Practices*, and *Soft Skills*. Table 1 contains a list of some of the different deficiencies identified in the systematic literature review.

Despite the wide availability of evidence to suggest the presence of several knowledge deficiencies, many of included studies did not provide much qualitative or descriptive information about the knowledge deficiencies that were reported. For example, a study by Lethbridge examined seventy-five different categories, including some such as data structures, databases, object-oriented concepts and technology, and several others where there was a large gap between what respondents indicated they had learned in college and what they needed for their present job [15]. While this information is highly useful, it does not provide additional insight into precisely what aspects of these categories need improvement. Software Testing, for instance, is one such category that is sufficiently broad that it is not immediately apparent where recent graduates may not be meeting employer expectations. In order to better understand some of these knowledge deficiencies, it is necessary to obtain better qualitative information about them.

Further, some of the studies identified in the literature review were published ten or more years ago, making the information dated [4,7,14,23]. The ACM/IEEE-CS joint task force have also released updates to their curricula recommendations for both computer science and software engineering between the publication of that research and the present time, which have likely had some impact on knowledge deficiencies [1,22]. This necessitates updated investigations into knowledge deficiencies in order to determine which are still prevalent, which have been remedied, and whether there are new knowledge deficiencies that have come into existence.

3. STUDY DESIGN

In order to validate the results found in the systematic literature review and those of previous researchers, and to gain additional understanding about problems that recently graduated students commonly face, the researchers of this study weighed several approaches to investigate knowledge deficiencies. It was decided that a semi-structured interview of industry managers would be the most effective method of obtaining data, as industry managers would have a broad perspective and a long history of dealing with knowledge deficiencies present in their new hires.

The remainder of this section provides a detailed description of the study, study participants, and data collection.

3.1 Research Questions

Before conducting the interviews, the authors of this study developed a series of research questions:

1. In what areas do recently graduated students most frequently struggle when beginning their careers in industry?

2. Are there certain skills, abilities, or knowledge (i.e. knowledge deficiencies) which recently graduated students might lack that can prevent them from gaining employment?

3. Are there issues that managers and hiring personnel attempt to identify during interviews with recent graduates that are still common after a recently graduated students begin their jobs?

3.2 Study Subjects

Participants in this study were twenty-three managers or hiring personnel at various companies located predominantly in the United States, but also included three participants from Europe. The companies ranged in size from small organizations employing under one hundred people to larger organizations with several thousand employees. Most of the participants either currently or previously worked for companies that had served as a sponsor company for the capstone course at NDSU.

Four of the participants worked as managers or team leads who were responsible for overseeing other employees. Eighteen participants worked as managers or team leads who

Table 1: Knowledge Deficiency Taxonomy

Soft Skills	Software Engineering Practices	Computer Science Concepts	Software Tools
Oral Communication	Design	Theoretical CS	Debuggers
Written Communication	Testing	Data Structures	Configuration Management
Leadership	Requirements	Programming	Development Tools
Presentation	Software Life-cycle	Networking	Misc. Software Tools

were also responsible for interviewing applicants for positions at their company. One of the participants worked in the company's human resources department. Although this person was primarily responsible for interviewing candidates at the company, they were still able to provide some feedback about issues that newly hired students experienced based on performance reviews and feedback from other managers.

3.3 Study Instrument

The instrument used in this study was a semi-structured interview. The semi-structured form of interview was chosen as it provided a good balance between the opportunity for participants to provide information about the most common problems experienced by new hires at their companies and for researchers to ask follow-up questions or get additional details from participants. As much of the prior literature did not provide precise details about the nature of knowledge deficiencies, it was a priority for the researchers to gain a more in-depth understanding of knowledge deficiencies. Hand-written notes were taken by the primary researcher during the interviews, and all interviews were conducted in English.

During the interview, participants were asked two primary questions. The first question related to the interview process at the company and what knowledge deficiencies prevented student applicants from obtaining a position at their company. The second was related to what issues or problems newly hired, recent graduates experienced while on the job or other skills, abilities, etc. that they had expected students to learn in college but were lacking. In some cases, interviewees would discuss topics that they felt were important for students to know. The interviewer attempted to determine whether or not the participant felt as though this was a problem at their company. If the interviewer was still unclear about a response it was not included in the results.

The interviewer would ask additional follow-up questions as necessary, usually for clarification or to gather additional details (e.g., "Is there a particular aspect of software testing where your new hires struggle most?"). In most cases, the interviewee would provide examples from the four broad categories (see table 1) described to them in the email (which was send when soliciting their participation in this research), but if the interviewer noticed that a particular category (e.g. software tools) had not been covered, a question about that category would be asked while avoiding going into further detail. This was done to prevent leading participants towards a particular response. Participants were also informed that they should not limit their responses to students who had graduated from NDSU. This was done to provide more generalized results, and in some cases the company had not hired any graduates from NDSU in the recent past or at all.

In a small number of cases, multiple managers from a single company were interviewed together. This occurred in some instances when the company had multiple, distinct departments or divisions. In these cases, the interviewer was careful to distinguish each participant's statements. This made it challenging in cases where one interviewee would speak about a particular problem and the other might make some indication that they had also noticed that problem. During two interviews which were conducted in-person, if the second interviewee would make a non-verbal indication of agreement, the interviewer would directly ask if they had also seen newly hired, recent graduates experience that issue.

3.4 Study Procedure

The study consisted of three main steps outlined below:

Step 1: Contact Potential Participants:

Potential participants were contacted via email and asked to participate in a brief interview related to the gap between the abilities of graduating students and the expectations of their company for new hires. Potential participants were provided a brief explanation of the purpose of the study and a general categorization of different areas where recently graduated students tended to have issues (i.e., personal skills, software tools, computer science concepts, and software engineering principles), and were given a brief list of examples, some of which can be seen in Table 1.

Step 2: Arrange Interviews:

Dates and times for an interview were arranged for those who indicated an interest in participating. In a few cases, a respondent indicated that they were not responsible for interviewing job applicants at their company or did not manage a group that contained recently graduated students, but that they could pass along the request someone at the company who met those criteria.

Step 3: Conduct Interviews:

Interviews were conducted primarily over the phone or via Skype, but in a small number of instances where the company was located in Fargo, North Dakota, the interview was conducted in person at the company's place of business. The interviews normally took between twenty and thirty minutes, but in a few cases the interview lasted nearly an hour.

4. RESULTS

Following the interviews, responses were grouped into a different categories, several of which had already been identified in previous research (see Table 1 for examples). However, some responses did not match closely with these existing categories. In these cases, a new category was created (e.g., code commenting). The researchers of this study were careful to distinguish between whether a problem was pri-

marily identified during the company's interview process or over the course of an employee's employment.

Table 2: Knowledge deficiencies identified in recently graduated students

Knowledge Deficiency	Interview	Job	Total
Project experience	13	0	13
Configuration management	0	12	12
Oral communication	9	2	10
Problem solving	8	3	10
Job Expectation	0	8	8
Testing	2	6	7
Databases	1	6	7
Teamwork	2	5	7
Working with customers	0	7	7
Written communication	1	6	6
Commenting code	0	6	6
Passion for tech. or job	4	2	6
Ability to see big picture	4	2	6

Table 2 shows the most frequent responses reported for problems that recently graduated students face when beginning their jobs or interviewing for a position at a company. These include a lack of project experience, difficulties using or unfamiliarity with configuration management tools, and problems with communicating effectively. Although there were several other common problems that were identified by multiple interview participants, the authors felt that it was best to focus on those responses which occurred most frequently. In some cases (e.g., software testing, oral communication, and problem solving abilities), an interviewee indicated that a particular issue was frequently a problem in both interviews with recently graduated students and when those students began their jobs. In that case, the issue is counted in both categories, but is only counted once in the total.

Additional qualitative information was added to provide further details (discussed in section 5) about each category. This was done to gain a better understanding of several categories, as some of the previous related work did not always provide descriptions for the categories or any further information to interpret the results [15]. This was beneficial as several categories such as software testing or design are fairly broad and can incorporate a large number of different aspects.

5. DISCUSSION

This section presents a discussion of the three research questions described in Section 3.1. Following the research questions, a brief comparison of these results with previous research is given.

5.1 Discussion of Research Questions

Research Question 1: In what areas do recently graduated students most frequently struggle when beginning their careers in industry?

The most frequent problem that recently graduated students experience is a proficiency in using (or knowledge of) different software tools. Of the twenty-three managers and hiring personnel interviewed, sixteen responded that their new hires commonly struggled with at least one software tool, and nine respondents indicated problems with multiple different types of software tools. Table 3 lists the number of times different categories of software tools were identified as tools with which recent graduates lacked proficiency.

Managers would frequently state that many of their new employees had not been exposed to these tools before, with configuration management tools being identified by over half of the respondents. Another common response was that students lacked experience setting up and using tools in a manner similar to an industry production environment. Furthermore, even if recent graduates were familiar with using a particular tool, several managers indicated that those graduates likely had not used it in a production environment before and lacked an inherent understanding of the importance of why a production environment should be used. One respondent reported that it could take up to six months for new employees to reach the same degree of proficiency as other employees with the software tools and systems used at the company. Other participants indicated that recent graduates only had a minimal or basic understanding of tools and one indicated that most had never used version control software beyond committing code to a repository and were incapable of merging or branching code. Multiple interviewees also indicated that most students did not have any exposure to continuous integration and regressing testing software such as Jenkins or Team Foundation Server. Two respondents also reported that recent graduates lacked proficiency with database management tools, with one indicating that those graduates lacked practical experience and did not know how to set up a new database.

Table 3: Software tools with which recently graduated students commonly lack proficiency

Software Tool	Times Identified
Configuration Management	12
Testing Tools	6
Integrated Developer Environment	5
Debugging Tools	4
Issue Tracking Tools	4
Code Analysis Tools	2
Database Management Tools	2

A lack of understanding of job expectations was the second most common problem that recent graduates were reported to experience on the job. One interviewee indicated that their new employees often seemed to be afraid of asking questions so as not to appear foolish and that they needed to understand that it was better to ask for help than to be stuck on something for extended periods of time. Another interviewee indicated that students had difficulty adapting to an eight hour work day after college where their work was split up over smaller segments. One respondent reported that many recent graduates lacked motivation and initiative and needed closer supervision in order to ensure that they remained productive. Multiple participants also indicated that recently graduated students occasionally lacked their professionalism. Some examples of this included dressing inappropriately or texting on their phones during meetings.

One respondent recounted having to deal with a new hire about personal hygiene issues.

Software testing was also frequently reported. Five of the six interview respondents specifically indicated that largest problem was related to the inability of recent graduates to construct good unit tests. One interviewee noted that recent graduates had a tendency to write redundant test cases and reintroduce old bugs. Another noted that they had to spend time training recent graduates on using their test platform (Gallio) and writing good unit tests. There were also four other respondents who had not identified problems related to testing itself, but had indicated that recent graduates lacked exposure to various testing tools such as continuous integration and regression testing tools.

Databases were another frequently reported issue. One interviewee reported that many of their new hires had trouble creating, designing, and interacting with databases. Another responded that recent graduates did not have a good understanding of the internal mechanics behind databases and that it could be difficult for them to learn that information when first starting their jobs. A third interviewee also indicated that new hires experienced problems when trying to tie in other code with databases. Two interviewees specifically pointed to a lack of experience with DBMS (database management systems) tools, with one indicating that recent graduates often did not even know how to use the tools to set up a new database.

Various aspects of communication were also identified. Written communication and the ability to work with customers were the most reported issues with which recent graduates struggled in their jobs, but oral communication was also reported by two respondents. One interviewee mentioned that recent graduates tend to have problems writing large memos or documents and have a tendency to communicate in small bursts, perhaps a side-effect of many youth communicating primarily through texting. Another respondent stated that students had difficulties communicating effectively with customers and tended to use too much technical jargon. Multiple interviewees indicated that many recent graduates had difficulty communicating with their bosses or asking questions if they were stuck or having issues. Another related aspect of this may be a lack of teamwork skills and collaborative ability, which was reported by five respondents, some of whom indicated that this was due to problems with communicating with their other team members. One respondent said that some recent graduates did not work well with others and struggled with successfully integrating with the rest of the team.

One final area where recent graduates were identified as struggling was code commenting. Two of the interviewed managers pointed out that comments from new hires often tend to be of poor quality, either offering no additional explanations or providing redundant information that is immediately obvious from the code. Two other respondents indicated that recent graduates often did not comment much if at all, and another stated that in addition to infrequent and poor comments, new hires did not initially follow the company coding standard which also caused problems.

Research Question 2: Are there certain skills, abilities, or knowledge which recently graduated students might lack that can prevent them from gaining employment?

Lack of project experience was the most frequently identified shortcoming that recent graduates faced when interviewing for a job. The majority of the interviewed managers indicated that they wanted applicants to have some kind of experience working on a large, team project. Several mentioned that having an internship or a co-op was something that they liked to see in applicants, but many mentioned that a senior capstone project was sufficient. One interviewee indicated that it was preferred if the applicant had worked individually on a large project. Multiple respondents also stated that they wanted student applicants to be able to explain their own role within a large team project and to describe how their work fit in with the rest of the project.

Table 4: Knowledge deficiencies identified during interviews with prospective student hires

Knowledge Deficiency	Times Identified
Project experience	13
Oral communication	9
Problem solving	8
Passion for tech. or job	4
Ability to see big picture	4
Teamwork	2
Continuous learning	2
Outgoing personality	2
Proactive self-management	2
Mobile development	2
Testing	2
Databases	1

Oral communication was another commonly identified problem for many recent graduates. Although multiple interviewees pointed out the importance of communication (one even stated that effective communication was perhaps even more important than good technical abilities), several also indicated that it wasn't always necessary to get a job. However, they also pointed out that those employees without effective communication skills were generally limited in advancement opportunities or the different positions in the company that they could occupy. Many of the interviewed managers responded that being able to articulate points and provide clear explanations during interviews were some of the main areas where some recent graduates had the most difficulty.

Although it was not identified as often as project experience or oral communication, many interviewees indicated that a lack of problem solving ability in a candidate was the most likely reason not to hire that person. One respondent stated that even if a student applicant did not know the answer to an interview question, they could still be hired as long as they could demonstrate a good problem solving approach. Another interviewee mentioned that it was common to give applicants small programming problems (e.g. develop an algorithm for searching a tree depth-first) to evaluate their problem solving ability. One interviewee also stated that it was important for applicants to be able to critically examine a problem and be able to ask good questions and display a deeper understanding of the problem.

When faced with several good applicants, some interviewees said that they tried to look for other personal qualities during the interview. The two most common were displaying a passion for technology or the field in which they would

be working and an ability to see the big picture. One respondent stated that it was important to find people who were passionate about their work, because even if they did not appear to be the most knowledgeable or skilled initially, that they could eventually be one of the better employees because of their drive and enthusiasm. Managers also mentioned several other important qualities such as leadership ability, having good collaborative skills, being proactive, displaying the ability to learn continuously, and having an outgoing personality. With few exceptions, most respondents indicated that these characteristics were not strictly necessary, but were important factors when determining who to hire and in many cases made more of a difference than technical ability.

However, there were some managers who indicated that there were certain technical abilities that were necessary. Testing ability, database knowledge, and mobile development experience were all reported. One respondent indicated that as part of the interview process, applicants were asked questions about how they would test various systems and were expected to be able to write unit tests for a small piece of code. Another stated that because databases were such an important part of the work done at that company that many student applicants had to be turned down simply because they did not have sufficient knowledge of or experience with databases. Respondents from two companies indicated that experience with mobile development was increasingly important as mobile devices such as smartphones and tablets become more prominent and that most of their new development was targeted at those devices.

Research Question 3: Are there issues that managers and hiring personnel attempt to identify during interviews with recent graduates that are still common after a recently graduated students begin their jobs?

In general, most of the interviewed managers indicated that the issues that they attempted to identify during interviews were not commonly present in recent graduates who had just started their working at the company. The results shown in Table 2 indicate that there is a clear difference between the knowledge deficiencies that managers identify in recent graduates during an interview for a position and the knowledge deficiencies that are seen in new hires beginning their first jobs in industry. Although there are some categories that overlap (e.g. problem solving, testing, teamwork, etc.) the majority of the identified knowledge deficiency categories are entirely, or almost entirely identified only during interviews (e.g. previous project experience) or only after a recently graduated student begins his or her job (e.g. configuration management) at the company. However, there were a few categories identified as being an issue both during interviews and on the job, which are listed in Table 5.

Table 5: Knowledge Deficiencies Existing in Both Interview Candidates and New Hires

Knowledge Deficiency	Times Identified
Oral Communication	1
Testing	1
Written Communication	1
Ability to Self-Manage	1
Problem Solving	1

Of the knowledge deficiencies identified, four were common issues identified either during interviews (oral communication and problem solving) or areas where recent graduates frequently struggled (testing and written communication) when starting their new job. One of the interviewed managers indicated that new hires often had difficult managing their own time and being proactive or taking initiative and that it was common for new hires to wait for additional instructors from their supervisor before beginning new tasks. The manager also indicated that they attempted to look for initiative during the interview process, but that it could be difficult to spot in candidates.

Other than those few instances, most interviewees did not report any overlap. One explanation for this is that many of the recent graduates who display the knowledge deficiencies most commonly identified during the interview process are far less likely to be employed at the company. It would also suggest the employers can effectively screen interview candidates based on whatever criteria they've established. Some of these deficiencies, such as oral communication and problem solving, were identified as problems that new hires experience on the job, indicating that not all companies evaluate interview candidates along those criteria or expect recent graduates to be proficient in those areas. One other possible explanation is that there are certain knowledge deficiencies (e.g. software tools) for which it may not be easy to gauge an applicants proficiency during an interview. However, there may be potential solutions to this problem. For example, one interview respondent, who mentioned that a working knowledge of Linux was necessary at their company, indicated that applicants were asked a few questions about various Linux commands in order to assess their level of knowledge.

5.2 Comparison with Previous Research

The systematic literature identified multiple knowledge deficiencies which were frequently identified by other researchers in previous studies [21]. The most frequently identified are listed in Table 6.

Table 6: Knowledge Deficiencies Frequently Identified in Literature

Knowledge Deficiency	# of Studies
Oral Communication	10
Written Communication	10
Testing	9
Project Management	9
Software Tools	9
Programming	7
Requirements	7
Teamwork	7
Problem Solving	7
Personal & Prof. Skills	6

Of the knowledge deficiencies frequently identified in the literature review, six were among the more frequently identified deficiencies by the managers and industry personnel interviewed in this study. Programming, one notable exception, was only identified by two interviewed managers in this study, and many interviewees made comments that re-

cent graduates usually had good programming skills. One of the studies identified in the literature review pointed to a lack of multi-threaded programming ability [18]. Another study indicated that language independence among recent graduates was a problem [20]. The results of this research did not indicate that programming ability was often viewed as a problem. Of the interviewees that mentioned programming in this study, one indicated that a good understanding of advanced object-oriented programming concepts was a common knowledge deficiency. Similarly, requirements were not frequently reported by interviewees in this study, with only a single respondent mentioning requirements. This person indicated that recent graduates struggled with working with customers to get specific requirements as well as precisely documenting requirements. Six other participants mentioned that recent graduates struggled when interacting with customers, but none specifically mentioned requirements gathering as a problem.

Similarly, project management was identified frequently in the literature review, but there was little additional information provided about which areas of project management needed the most improvement [8, 9, 14] One study did report that software cost estimation was one aspect of project management was an area where recent graduates underperformed in relation to industry expectations [16]. Two of the managers interviewed in this study reported issues with project management. One reported that newly hired graduates struggled to break down tasks and estimate size and time for them, familiarity with using burn down charts to track progress, and account for external complications (and other issues related to risk management). The other reported that recent graduates commonly lacked the ability to plan ahead and frequently possessed poor estimation abilities.

Testing was one of the most frequently identified knowledge deficiency in the literature review. Several of the papers did not provide an any details of which areas of testing were the problem [9,14,15]. However, some did provide additional details. One study reported that senior-level students lacked the ability to effectively use testing tools and that they often produced test suites with a large number of redundant tests [5]. The results from this study indicate that writing unit tests and using software tools for continuous integration and regression testing are the largest areas where recent graduates are lacking in terms of testing. One respondent also indicated that recent graduates often did not have any experience (or even knowledge of) test-driven development.

In regards to soft skills, one study indicated that experience working as part of a team was necessary [20]. Another study reported that students didn't receive much experience working as part of a cross-disciplinary team in college, but that these types of teams were common in industry [24]. Another study reported that recent graduates had communication issues, primarily focused on communicating when they were struggling and in need of assistance from a coworker or manager [3]. Other studies also reported issues such as a lack of motivation, ability to accept criticism, and the ability to listen and pay attention [18,27]. The results of this study were similar, with the most common responses indicating that new hires had difficulties asking for help when stuck on something, and that new hires had issues communicating with other members of their teams. There was also an overlap in issues with personal and professional skills, where recent graduates do not always display the appropriate behavior that's expected of them.

Although problem solving was frequently identified, many of the studies did not provide additional details. One study stated that new hires often lacked the ability to generate alternative solutions to a given problem [7]. The results from this study indicate that problem solving was considered an important skill, perhaps the most important, when interviewing recent graduates applying for jobs. Both the importance of demonstrating a sound approach to problem solving and the ability to critically examine a problem and ask important questions about that problem were identified as important qualities for applicants to display during interviews with perspective employers.

That the results of this study have several similarities with previous research provides support to validity of these results. However, that some knowledge deficiencies that were frequently identified in the literature (e.g., programming) were not identified with great frequency in this research, and that others which were not generally identified in prior research were identified frequently in this study (e.g., project experience) indicates that these or other results are not exhaustive. It may also be indicative of shifts in the importance placed on different areas within industry over time as the literature review incorporated publications from the mid-90's onwards. Another possible explanation is that some regional differences may exist, although the similarity of the results does indicate that they are generalizable to the entire industry.

5.3 Other Discussion

There also exists some debate as to what academia should do with this information. There are those who feel strongly that college should prepare students for their future careers and should incorporate this information into curriculum design and teaching practices. There are also those who are opposed to the idea and feel as though college should not be treated as a vocational school. The first group would argue that unless their education prepares students for their careers, it does not have value and will hinder students in achieving success in industry. The second group would argue that teaching students a specific set of skills needed by industry would only limit their career choices and that it's better to have a wider education, even if some of it may not be useful for many jobs. Both arguments have merit, but this research does not seek to suggest that either point of view should be followed.

For obvious reasons it would not be practical to prepare students to be proficient in every tool, technique, etc. used by all of the different industry companies. However, there are shortcomings among graduating students that can easily be addressed by educators. Some of these issues can be addressed through curriculum changes, whereas others may be best dealt with by specifically educating students about the existence of existing knowledge deficiencies. The overall purpose of this research is to provide educators with a better understanding of which areas are most in need of improvement and to allow them to determine what, if any, action should be taken.

Some of the most frequently identified knowledge deficiencies can be mitigated without much difficulty. Some such as heaving "real-world" project experience would require adding a capstone project course or requiring that

students complete a co-op with some company before graduation, but many such as unfamiliarity with configuration management, unit testing, and other software tools can be addressed by introducing these tools early in the curriculum and having instructors incorporate their use into multiple courses to ensure that students have at least a moderate familiarity with these tools. Other practices such as commenting code that were frequently identified in this study could be similarly addressed by ensuring that beginning students began to incorporate them into their work. Other knowledge deficiencies such as problem solving may require a more careful examination of the curriculum to determine if students are receiving adequate levels of instruction. Some knowledge deficiencies such as job expectations may be difficult to incorporate into a course work. A project-based capstone course or an industry co-op may be the best solution, but not all universities have those requirements or can easily add them to their program. Adding either of those may help to provide students with a better understanding of working in industry as well as developing teamwork skills

6. THREATS TO VALIDITY

Because the interviewed managers and hiring personnel were employed at companies predominantly located in the United States, there is the possibility that the results can not be generalized to all software development companies throughout the world. The business areas of the different companies were varied enough to suggest that the results are not too specific to any particular business domain (e.g. banking software, the medical field, etc.) or could not be generalized to software development roles in general. The sizes of the companies were also similarly varied from large companies employing hundreds of developers to smaller organizations with tens of developers. Additionally, the researchers were careful to notify interviewees that their responses should not be limited to knowledge deficiencies among students from the researchers' university in order to ensure that the results were not limited to students or the curriculum from one university. That the results of this study are in-line with the findings from previous studies would also help to suggest that the results are generalizable.

None of the interviews were recorded, which in some instances lead to small ambiguities or vagueness about some of the identified problems. In a few cases, more detailed information was not provided, either because the interviewee did not provide additional details or because the interviewer was not able to record all of the details of the conversation. In this case, the identified problems were still counted as being identified, but no additional qualitative information was reported.

Because study participants were not asked about specific problems experienced by recently graduated students from a compiled list, there is a possibility that some issues are under-represented in the results of the study. However, this was considered preferable to specifically asking about particular skills, abilities, etc. and potentially biasing the results. There is also the possibility that areas in which recently graduated students struggle in their new jobs were not identified if the interviewed manager had expectations that their new hires would struggle in that area and merely expected them to learn this part of their job. Also, because not every company requires the same skills of their developers, it is not possible to treat all categories equally. For example, if a company does not write software which interacts with databases, it is unlikely that they will require applicants with database knowledge or report issues related to new employees' abilities to use databases. Because participants were not specifically asked about all of the previously identified knowledge deficiencies, it is possible that the results may be slightly skewed and undervalue the importance of some knowledge deficiencies.

7. CONCLUSION AND FUTURE WORK

Previous research has identified various issues that recently graduated computer science students struggle with when starting their new jobs. This paper provides additional support for the existence of several of these knowledge deficiencies including the use of configuration management tools, communication skills, and testing ability. This research also provides additional qualitative information for knowledge deficiency categories that may not have been well defined or contained additional details in prior research and provides a more recent appraisal of knowledge deficiencies [14].

The most commonly reported knowledge deficiencies that educators should address involve providing some type of real-world project experience, ensuring that students are exposed to the tools most commonly used in the software industry, making sure that students have effective problem solving skills and that students can effectively communicate their solutions to others, and that students have experience with writing unit tests for their code. Although it is not something that can be easily incorporated into most courses, effort should also be taken to ensure that students have some understanding of the expectations of working in industry and working with actual customers and clients.

Further research in this area includes interviewing managers in other countries (e.g., India, China, etc.) in order to gain insight into how knowledge deficiencies vary across regions outside of the United States. It would also be beneficial to approach this research from other angles, such as by surveying or interviewing software developers who have begun employment within the last few years as they would be able to provide further information about knowledge deficiencies. Another potential approach would be working with companies to conduct studies of senior-level computer science students to determine if their education has prepared them for industry.

8. REFERENCES

[1] J. M. Atlee, R. J. LeBlanc, Jr., T. C. Lethbridge, A. Sobel, and J. B. Thompson. Software engineering 2004: Acm/ieee-cs guidelines for undergraduate programs in software engineering. In *Proceedings of the 27th international conference on Software engineering*, ICSE '05, pages 623–624, New York, NY, USA, 2005. ACM.

[2] A. Begel and B. Simon. Novice software developers, all over again. In *Proceedings of the Fourth international Workshop on Computing Education Research*, ICER '08, pages 3–14, New York, NY, USA, 2008. ACM.

[3] A. Begel and B. Simon. Struggles of new college graduates in their first software development job. In *Proceedings of the 39th SIGCSE technical symposium on Computer science education*, SIGCSE '08, pages 226–230, New York, NY, USA, 2008. ACM.

[4] D. J. Byrne and J. L. Moore. A comparison between the recommendations of computing curriculum 1991 and the views of software development managers in ireland. *Comput. Educ.*, 28(3):145–154, Apr. 1997.

[5] J. Carver and N. Kraft. Evaluating the testing ability of senior-level computer science students. In *Software Engineering Education and Training (CSEE T), 2011 24th IEEE-CS Conference on*, pages 169 –178, may 2011.

[6] A. Eckerdal, R. McCartney, J. E. Moström, M. Ratcliffe, and C. Zander. Can graduating students design software systems? In *Proceedings of the 37th SIGCSE technical symposium on Computer science education*, SIGCSE '06, pages 403–407, New York, NY, USA, 2006. ACM.

[7] H. Haddad. Post-graduate assessment of cs students: experience and position paper. *J. Comput. Sci. Coll.*, 18(2):189–197, Dec. 2002.

[8] D. Hagan. Employer satisfaction with ict graduates. In *Proceedings of the Sixth Australasian Conference on Computing Education - Volume 30*, ACE '04, pages 119–123, Darlinghurst, Australia, Australia, 2004. Australian Computer Society, Inc.

[9] B. Kitchenham, D. Budgen, P. Brereton, and P. Woodall. An investigation of software engineering curricula. *Journal of Systems and Software*, 74(3):325 – 335, 2005.

[10] B. Kitchenham and S. Charters. Guidelines for performing Systematic Literature Reviews in Software Engineering. Technical Report EBSE 2007-001, Keele University and Durham University Joint Report, 2007.

[11] B. A. Kitchenham. Procedures for undertaking systematic reviews. Technical report, Computer Science Department, Keele University, 2004.

[12] D. Knudson and A. Radermacher. Software engineering and project management in cs projects vs. "real-world" projects: A case study. In *Proceedings of the IASTED Internation Conference on Software Engineering and Applications*, SEA '09, 2009.

[13] D. Knudson and A. Radermacher. Updating cs capstone projects to incorporate new agile methodologies used in industry. In *Proceedings of the 2011 24th IEEE-CS Conference on Software Engineering Education and Training*, CSEET '11, pages 444–448, Washington, DC, USA, 2011. IEEE Computer Society.

[14] T. C. Lethbridge. A survey of the relevance of computer science and software engineering education. In *Proceedings of the 11th Conference on Software Engineering Education and Training*, CSEET '98, pages 0056–, Washington, DC, USA, 1998. IEEE Computer Society.

[15] T. C. Lethbridge. Priorities for the education and training of software engineers. *J. Syst. Softw.*, 53(1):53–71, July 2000.

[16] T. C. Lethbridge. What knowledge is important to a software professional? *Computer*, 33(5):44 –50, may 2000.

[17] C. Loftus, L. Thomas, and C. Zander. Can graduating students design: revisited. In *Proceedings of the 42nd ACM technical symposium on Computer science education*, SIGCSE '11, pages 105–110, New York, NY, USA, 2011. ACM.

[18] M. M. McGill. Defining the expectation gap: a comparison of industry needs and existing game development curriculum. In *Proceedings of the 4th International Conference on Foundations of Digital Games*, FDG '09, pages 129–136, New York, NY, USA, 2009. ACM.

[19] E. G. McGuire and K. A. Randall. Process improvement competencies for is professionals: a survey of perceived needs. In *Proceedings of the 1998 ACM SIGCPR conference on Computer personnel research*, SIGCPR '98, pages 1–8, New York, NY, USA, 1998. ACM.

[20] C. S. Miller and L. Dettori. Employers' perspectives on it learning outcomes. In *Proceedings of the 9th ACM SIGITE conference on Information technology education*, SIGITE '08, pages 213–218, New York, NY, USA, 2008. ACM.

[21] A. Radermacher. Evaluating the gap between the skills and abilities of graduating computer science students and the expectation of industry. Master's thesis, North Dakota State University, 2012.

[22] M. Sahami, M. Guzdial, A. McGettrick, and S. Roach. Setting the stage for computing curricula 2013: computer science – report from the acm/ieee-cs joint task force. In *Proceedings of the 42nd ACM technical symposium on Computer science education*, SIGCSE '11, pages 161–162, New York, NY, USA, 2011. ACM.

[23] E. Scott, R. Alger, S. Pequeno, and N. Sessions. The skills gap observed between is graduates and the systems development industry - a south african experience. In *Proceedings of the Informing Science and IT Education Conference*, IS2002, pages 1403–1411. Informing Science, june 2002.

[24] G. Scott and D. N. Wilson. Tracking and profiling successful it graduates: An exploratory study. In *Proceedings of the 13th Australasian Conference on Information Systems*, ACIS '02, pages 1185–1195, 2002.

[25] C. B. Simmons and L. L. Simmons. Gaps in the computer science curriculum: an exploratory study of industry professionals. *J. Comput. Sci. Coll.*, 25(5):60–65, May 2010.

[26] L. A. Sudol and C. Jaspan. Analyzing the strength of undergraduate misconceptions about software engineering. In *Proceedings of the Sixth international workshop on Computing education research*, ICER '10, pages 31–40, New York, NY, USA, 2010. ACM.

[27] D. B. Tesch, G. F. Braun, and E. A. Crable. An examination of employers' perceptions and expectations of is entry-level personal and interpersonal skills. *Information Systems Education Journal*, 6(1):3–16, 2008.

[28] E. M. Trauth, D. W. Farwell, and D. Lee. The is expectation gap: industry expectations versus academic preparation. *MIS Q.*, 17(3):293–307, Sept. 1993.