The role of anxiety when learning to program: A systematic review of the literature

Keith Nolan
Department of Computer Science,
Maynooth University,
Ireland
keith.nolan@nuim.ie

Susan Bergin
Department of Computer Science,
Maynooth University,
Ireland
susan.bergin@nuim.ie

ABSTRACT

According to the World Health Organisation the number one global health issue for young people is their mental health. For students, mental well-being is associated with effective learning, and their ability to navigate through university/college, having the resilience to cope with the challenges and stresses of student life.

In Ireland, Computer Science (CS) non-progression rates are alarming, with a large number of students failing to progress each year. Currently non-progression rates are 25% in CS, significantly higher than the national average of 16% across all other fields of study. On top of the normal stressors of transitioning or returning to university, CS students are arguably exposed to a unique set of factors that could further induce anxiety. First, they typically have no formal CS exposure or training to draw on. Second, the number of contact hours and workload are high. Third, CS courses includes programming modules. For some, learning to program is difficult and many struggle to master the core concepts. Learning typically takes place in a lab environment where inexperienced programmers will begin to type ("code") shortly after being presented with a problem rather than spending time designing a solution. Thus the lab becomes active and busy from the onset, making struggling students cripplingly perceive their peers know more. Further, novice programmers use the compiler to constantly monitor their progress and error messages can be perceived as negative feedback. Such an environment can create or compound anxiety and stress. At our institution a large number of CS students register for counselling services or leave.

In this paper we present a systematic literature review on the role of anxiety when learning to program. The work is novel, valuable, and timely. The approach used is systematic, in that a structured search of electronic resources has been conducted and the results are presented and quantitatively analysed. A detailed discussion on the findings is provided and important implications for the teaching and learning of programming are described.

Koli Calling 2016, November 24-27, 2016, Koli, Finland
© 2016 ACM. ISBN 978-1-4503-4770-9/16/11…$15.00
DOI: http://dx.doi.org/10.1145/2999541.2999557

CCS Concepts

•Social and professional topics → Computing education;

Keywords

Anxiety; learning; programming; computer science; systematic review

1. INTRODUCTION AND MOTIVATION

The mental health of young people is an increasing concern worldwide [17]. In 2012, a study (My World Survey), was carried out to investigate the mental health of over 8000 young adults (aged 18 - 25) in Ireland, the majority of whom were attending tertiary education [11]. Of concern, the study found that in any group of 100, 40 suffer from depression and 38 from anxiety with the three main stressors cited as college, money and work.

Computer Science (CS) non-progression rates in Ireland are currently 25%, significantly higher than the national average of all disiplines of 16% [24]. This problem is not unique to Ireland but is mirrored in many other countries as well [43]. Students who take CS at third level typically have no previous formal exposure to the subject. CS courses are often coupled with other subjects such as Mathematics, Engineering and other Science subjects. The workload in this type of course (weekly lectures, assignments, tutorials and labs) conceivably adds to the stress of tertiary life. In addition studying CS can be a solitary experience [1]. Many of the assignments involve working alone at a computer. This is not unique to CS courses but it is echoed in industry where CS/IT specialists often work alone. Empirical evidence suggests the depressive states of such working environments and in particular of the associated high workload [29]. Furthermore, Thomèe reported the negative implications of technology on mental health [39], including sleep disturbances and symptoms of depression.

CS courses require students to undertake programming modules. Learning to program is difficult and results in high failure rates worldwide [4]. Since the 1970's, a plethora of research has taken place investigating how students learn to program and the factors that influence a student's likelihood of success. In Maynooth University, numerous initiatives have been introduced to support the study of programming such as Problem Based Learning, the use of novel teaching methods such as reprogrammable robots and a Programming Support Centre [31]. These initiatives have improved programming performance with various degrees of success. Additionally, the Computer Science Education research group at Maynooth University has been conducting research on factors that influence student performance for over a decade. This work has led to the development of an automated web based computational model known as Press# (**P**redict **S**tudent **S**uccess Sharply) [33]. This model can predict the performance of novice programmers after minimal exposure to programming concepts with 80% accuracy, irrespective of the language being studied, the student cohort (major/minor, university, community college etc.) or other aspects of student profile (age, gender etc.). Over the past 10 years the system has been repeatedly re-validated with students in Ireland and abroad and it continues to perform with such high accuracy. Interestingly, the most significant factor in the model is "self-efficacy", a student's judgement of their ability to program [5]. This perception is likely linked to components of a students mental health, for example low levels of self-efficacy are often associated with anxiety [12, 21].

For many, tertiary level introductory programming modules are their first experience of programming. Learning to program is a public endeavour where a student can perceive they are struggling even if this is not the case. Typically, students learn to program in a lab with potentially a large number of peers present. For some, this can become a source of anxiety - presuming that you know less than your peers based on the busyness of the lab, that is, other learners "coding up" solutions when you don't know where to start (although the "busy" students may not know any more). Coupled with this, programming is binary in terms of correctness - a program will either compile or it won't. When you are a novice programmer, receiving a long list of compiler errors can be perceived as a form of negative feedback which could induce or increase anxiety and ultimately lead you to perceive that you are failing at learning to program.

Over the past fifteen years, at labs, lectures, tutorials and at the Programming Support Centre we have witnessed first-hand many anxious and stressed students. This, coupled with the findings in the *My World Survey*, the relationship between anxiety and self-efficacy along with the specifics of learning to program (new subject, noisy lab etc.) justifies a comprehensive evaluation of anxiety when learning to program.

For clarity, anxiety is defined as a "feeling of worry, nervousness or unease about something with an uncertain outcome" [37]. More specifically, anxiety can be split into two categories: state anxiety and trait anxiety. State anxiety is the unpleasant feelings when confronted with specific situations, demands or even particular objects [7]. Trait anxiety refers to the stable tendency to attend to, experience, and report negative emotions such as fears, worries, and anxiety across many situations [20]. We are interested in all aspects of anxiety given its implications for learning [36].

As a first step we extrapolated further on the potential causes of anxiety in our students. Obviously programming as a subject (language, syntax, error generation, learning environment etc.) is important, test anxiety is relevant (as it is to every discipline), computer anxiety is important (given the volume of computer usage) and potentially anxiety associated with mathematics given its strong relationship with CS and programming (e.g. writing programs to determine prime numbers, greatest common divisor, factorial, etc).

To this end our goals are two-fold:

1. With mental health becoming an increasing concern there is a clear need to carry out a review of the role of anxiety of students when learning to program by considering anxiety associated with programming itself, mathematical concepts, computer usage and assessment.

2. To provide a *systematic review* of the literature as an example of how this process is implemented and highlight its value to the Computer Science Education Research community.

2. RESEARCH QUESTIONS

The starting point of this review was to develop our research questions. To this end, we defined several research questions to incorporate the likely breath of sources of anxiety when learning to program. The defined questions are as follows:

1. Is there a relationship between learning to program (language, syntax, compilation etc.) and anxiety?

2. Is there a relationship between mathematical anxiety and learning to program?

3. Does computer usage cause anxiety when learning to program?

4. Does test anxiety affect programming and more broadly Computer Science students?

3. METHOD

3.1 Introduction

This systematic literature review is based on Kitchenham's method as applied to software engineering [23]. This method of performing a review was chosen as the process is well documented and is derived from review processes that were previously well established in the medical community. Kitchenham outlines how to identify the need for the review, how to develop a strict protocol to follow for the review and how to report the findings from the review.

The following steps are listed in the method:

- Identify the need for a systematic literature review and define your research questions.
  Addressed in Section 1 and Section 2.

- Carry out an exhaustive search for studies.
  Discussed in Section 3.2 and Section 3.3.

- Assess quality of accepted studies .
  Discussed in Section 3.5.

- Extract data from accepted studies.
  Discussed in Section 3.6.

- Compile background information on the studies.
  Discussed in Section 4.

- Summarise and synthesise study results.
  Discussed in Section 5.

The correct application of these steps leads to a rigorous, exhaustive and reproducible meta-review [23].

3.2 Search Terms

In this study two primary search terms were used: *Anxiety* and *Programming.*

The words *"emotional arousal"* and *"stress"* were also used in place of *"anxiety"* to increase the search scope. In addition to the above search terms, the following secondary search terms were used to narrow the number of results returned from the databases:
Learning, *Mathematics*, *Computer*, *Test (Exam).*

3.3 Resources Searched

An extensive search of five publication repositories was carried out between February and July 2016 using the search terms mentioned in Section 3.2. The repositories were: ACM Digital Library, IEEE Xplore, ERIC, Science Direct and Google Scholar.

The ACM Digital Library (ACM DL) contains over 430,000 full text papers. When searching the ACM DL with the primary search terms, "anxiety" returned 271 results and "programming" returned 96,447 results. As secondary search terms were added, the number of results returned further decreased. It was decided that all papers returned in the "anxiety" search would be screened due to the relatively small number.

The IEEE Xplore database contains three million citations. It was searched using the same search criteria as the ACM DL. As there were only 43 results returned after searching "anxiety" and "programming", all papers were screened.

The ERIC database was then searched as the database is specifically for papers relating to education. We used the same search criteria as used for the ACM DL and IEEE Xplore. The search only found papers that had previously been found in either the ACM DL or IEEE Xplore. Science Direct contains over 12 million citations relating to Physical Sciences and Engineering, Life Sciences, Health Sciences, and Social Sciences and Humanities. The database was searched to identify any other research related to our research questions, using the same search criteria. No additional sources were found. Google Scholar was used as a final search space to eliminate the likelihood that a relevant publication had been missed. No additional studies were found.

3.4 Document Selection

From searching the databases and referenced material, a total of 84 studies were identified on title alone to address the research questions. Full texts of those studies were then obtained. The abstracts for all 84 unique studies were then screened to exclude any studies that were not directly related to the research questions.

Inclusion and exclusion criteria were developed as recommended in the systematic review guidelines [23].

Texts were included that:

- directly answered one or more research question.
- focused on anxiety in programming.
- focused on anxiety which related to either mathematics anxiety, computer anxiety or test anxiety.

Studies were excluded that:

- were in the form of a book or grey literature (opinion pieces, technical reports, blogs, presentation, etc.).
- related to primary or secondary school learning (one study was kept as it was deemed relevant [22]).

3.5 Quality assessment

Each primary study was evaluated based on quality assessment criteria defined in Kitchenham's systematic literature reviews for software engineering [23]. The most relevant questions were taken from a set of 18 questions and applied to this review. These questions were:

- How credible are the findings?
- How well does the evaluation address its original aims and objectives?
- Hew well was the data collection carried out?
- How well can the route to any conclusions be seen?
- How adequately has the research process been documented?

The five questions were the only questions which were deemed to give the greatest overview of the selected studies. A scoring system was developed in order to grade each of the studies.

- Question 1 : Y (yes), the findings are very credible, P (partly), the findings are partially credible, N (no), the findings are not credible.
- Question 2 : Y (yes), the evaluation addresses the original aims and objectives, P (partly), the evaluation addresses the original aims and objectives implicit, N (no), the evaluation does not address the original aims and objectives.
- Question 3 : Y (yes), the data collection was carried out well and outlined clearly, P (partly), the data collection was carried out well but not outlined clearly, N (no), the data collection was not carried out well.
- Question 4 : Y (yes), the route to the conclusion is clearly seen, P (partly), the route to the conclusion is implicit, N (no), the route to the conclusion can not be inferred.
- Question 5 : Y (yes), the research method is well documented, P (partly), the research method is implicit, N (no), the research method can not be inferred.

The scoring was Y = 1.0, P = 0.5, N = 0.0. The threshold for an accepted study was 3.0 The score for each study in shown in Table 1.

Table 1: Quality assessment of studies

Reference	Q1	Q2	Q3	Q4	Q5	Total
[2]	Y	P	Y	P	P	3.5
[6]	Y	P	Y	Y	P	4
[26]	Y	Y	Y	Y	P	4.5
[10]	Y	Y	Y	Y	Y	5
[12]	P	P	Y	P	P	3
[8]	Y	Y	Y	Y	Y	5
[13]	Y	Y	Y	Y	Y	5
[22]	Y	Y	Y	Y	Y	5
[25]	Y	Y	Y	Y	P	4.5
[7]	Y	Y	Y	Y	P	4.5
[35]	Y	Y	Y	P	Y	4.5
[40]	Y	Y	Y	Y	Y	5
[9]	Y	Y	Y	Y	Y	5
[15]	P	Y	Y	Y	Y	4.5
[16]	Y	Y	Y	Y	Y	5
[18]	Y	Y	Y	Y	Y	5
[19]	Y	Y	Y	Y	Y	5
[14]	Y	Y	Y	Y	Y	5
[27]	Y	Y	Y	Y	Y	5
[28]	Y	Y	N	Y	Y	4
[30]	Y	Y	Y	Y	Y	5
[38]	Y	Y	Y	Y	Y	5
[32]	P	Y	Y	P	P	3.5
[42]	Y	Y	Y	Y	Y	5

3.6 Data Extraction and Synthesis

From the 84 studies found based on title, 60 of those were subsequently rejected after reading the abstract. The remaining papers were screened using our inclusion/exclusion criteria. A final list consisting of 24 relevant papers that satisfied the inclusion criteria and informed the research questions was compiled.

Mendeley reference manager was used to record the reference details of each study. Along with this, a separate document was used to record additional results that Mendeley couldn't include e.g. brief summary of the study. Extracted data from the 24 studies is given in Table 2.

4. RESULTS - BACKGROUND

4.1 Types of studies

Of the 24 papers accepted, 83% of the studies were empirical studies and interviews. These studies were evidence based studies where data was collected largely by questionnaires, however, some were experimental. Literature reviews on computer anxiety accounted for 17% of the accepted papers.

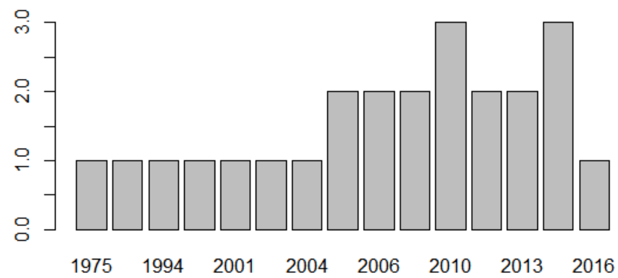
4.2 Temporal view of publications

The distribution of the primary studies throughout the years is shown in Figure 1. As can be seen there is an increase in the number of publications since 2005, showing a growing interest in the research area.

4.3 Data sources

All studies chosen for this review were either published in conference proceedings or journals. Table 2 shows the distribution of primary studies derived from their publication channels.

Figure 1: Number of papers collected each year



5. RESULTS

This section reports the findings from the literature review for each of the research questions. Although an extensive review was carried out, only 24 studies were found to address the research questions. Some of these studies were only useful for one question while others addressed more than one research question.

Although the number of studies is not large they do make valuable contributions in that they either have a large number of participants or are longitudinal in nature or have interesting implications for the knock on effect of anxiety in learning to program over time.

5.1 Is there a relationship between learning to program (language, syntax, compilation, etc.) and anxiety?

This research question was informed by nine studies: Connolly et al. [8], Guynes [18], Chang [6], Scott et al. [35], Melin et al. [27], Falkner et al. [13], Gerritsen et al.[16], Hamer et al.[19] and Ngai et al. [30].

Connolly et al. through a longitudinal study investigated anxiety when studying CS. Specifically, the study investigated the variance of anxiety amongst undergraduate computing students, with an emphasis on learning to program during their first year. This study was conducted over two years and 86 students participated. The study was set up in two parts, where students took a questionnaire at the start of their first year and then again at the end of first year. From the study, two important factors were investigated: 1) computer self concept and 2) state of anxiety.

Computer self concept

Computer self concept looked at a student's self-efficacy or confidence. It was measured across 11 questions. In the pre-survey, 23% of students claimed they were *"unsure"* when asked if they would be able to learn a programming language. This was before they had any experience on the course. Such a finding is perhaps understandable in that programming is a new subject but a cause of concern given that people already have a negative perception about/related to their programming ability.

State of anxiety

This category examined the cognitive, emotional and psychological states of anxiety that students face in programming situations. Before the semester began, 44% of students reported that they did not feel relaxed when using a computer let alone programming. However, it was noted that the students sense of worry did diminish by the end of the

Table 2: Accepted studies

Reference	Keywords	Publication Source	Type[1]	Type of study[2]	Number of participants
Baloglu et al.[2]	College students, mathematics anxiety, multivariate differences	Personality and Individual Differences	J	Q	759
Chang [6]	Computer anxiety, computer experience, path analysis, task complexity	Computers in Human Behaviour	J	Q	307
Maurer[26]	Meta-analysis, computer anxiety, computer experience	Computers in Human Behaviour	J	LR	n/a
Deloatch et al.[10]	Computer-based testing , programming-centric, test anxiety	SIGCSE	C	Q	391
Doyle et al.[12]	Computer anxiety, computer experience, computer science, self-efficacy	Frontiers in Education	C	Q	163
Connolly et al.[8]	Longitudinal research, programming anxiety	Transactions in Education	C	Q	86
Falkner et al.[13]	Collaborative activities	SIGCSE	C	I	10
Kavakci et al.[22]	Social anxiety, test anxiety	Dusunen Adam	J	Q	436
Macher et al.[25]	Mathematical anxiety, academic succession	Eur J Psychol Educ	J	Q	147
Chua et al.[7]	Computer anxiety, computer experience	Computers in Human Behaviour	J	LR	n/a
Scott et al.[35]	Anxiety, self-belief, self-efficacy	ICER	C	Q	239
Todman et al.[40]	Psychological genders, computer anxiety	Computers in Human Behaviour	J	Q	138
DeRaadt [9]	Anxiety, cheat sheet	ACE	C	E	89
Fone [15]	Computer anxiety, neural networks	Neural Networks	C	E	21
Gerritsen et al.[16]	Stressful programming	WI-IAT	C	Q	21
Guynes [18]	Computer anxiety, system response time	Communications of the ACM	J	E	93
Hamer et al.[19]	Peer assessment, anxiety	ICER	C	Q	1500
Fenwick et al.[14]	Test anxiety, computer science	SIGCSE	C	Q	100
Melin et al.[27]	Project work, collaborative learning, anxiety	ITICSE	C	Q	60
Mills[28]	Program correctness	ACM	J	LR	n/a
Ngai et al.[30]	Self-assessment, anxiety	SIGCSE	J	Q	13
Suraweera [38]	Anxiety, mathematics, computer science	ACM	J	LR	n/a
Owolabi et al.[32]	Programming anxiety, mathematical anxiety	GSTF Journal on Computing	J	E	160
Vitasari et al.[41]	Anxiety, academic performance.	Procedia - Social and Behavioural Sciences	J	Q	205

[1]C=Conference,J=Journal
[2]Q=Questionnaire, LR=Literature Review, I=Interview, E=Experimental

year.

One of the key findings is the students perception of their ability to learn how to program. Connolly gathered feedback from the participants of the study and found evidence that novice programmers were computer phobic, for example *"I'm afraid I'll wreck the program/hard drive"*. For novice programmers, receiving any sort of programming error can be a source of displeasure which could lead to anxiety. Coupled with this, Connolly et al. also found that confidence and self-efficacy affects learning to program. It was found that the lower the confidence and self-efficacy, the harder it was for a student to complete a programming task correctly.

Guynes investigated the impact of system response time on state anxiety. Eighty six participants took part in the experiment in which they had to edit a file containing 28 errors. In their analysis, Guynes reported that there was a statistically significant relationship between state anxiety and system response time (α=0.05,p=0.0155) [18].

Chang investigated if there was a relationship between anxiety and programming-task complexity and how this relates to programming skills [6]. The study consisted of 307 participants and measured perceived task complexity and self-reported anxiety levels using the Computer Attitude Scale. Results were based on three different levels of programming task complexity from easy to hard. Results showed that there was a significant relationship between perceived programming task complexity and anxiety levels, that is, as perceived programming task complexity increased, so did perceived anxiety levels.

Scott et al. hypothesised that students programming practice behaviour is negatively impacted by anxiety [35]. The original intention of Scott et al.'s study was to assemble and validate an instrument to assess self-belief in CS1. Two hundred and thirty-nine students participated in this study. Of note, they found in terms of programming anxiety students often worry when completing debugging tasks and they would start to feel nervous when they try to find and fix programming bugs.

Melin et al. investigated how project orientated work effects learning [27]. The project orientated work was incorporated into the course. A total of 60 CS students participated in the course over a 15 week period. The biggest worry for students was that their grade would be affected by other students who didn't pull their weight. Students worries were elevated by the introduction of a clear marking scheme. By the end of the course students felt more confident about their programming skills.

Group work is becoming more popular in programming. With group work, a students work is constantly being scrutinised by peers. While the students work is not being formally assessed, the fact that their peers are assessing the work can cause anxiety. Falkner et al investigated how collaborative activities may introduce stress and anxiety for students [14]. In their study 10 students participated in an interview. The goal of the interview was to understand from the students perspective 1) the purpose of collaborative activities, 2) whether collaborative activities are perceived as positive or negative experiences and 3) how relationships between students within the groups worked. They concluded that students were stressed and anxious when working in groups. This is due to students not working as a group but rather as individuals when completing tasks.

Gerritsen et al. investigated the effects that pressure and stress can have on a learner [16]. In their study they investigated physiological signals during high stress activities. They had a total of 21 participants and found that during high stress moments, the perception of the complexity of the task can define how hard a task actually is for that person.

Hamer et al. reported on a large scale scale study of 1500 students on the topic of peer assessment [19]. They reported that peer assessment is a source of anxiety to students as the mark received from different peers may be vastly different depending on the relationship to those peers.

Ngai et al. conducted a study which aimed to see if self assessment helps to reduce student frustration and anxiety. Thirteen participants took in the study. The participants were asked to i) grade their own ability level and ii) self-grade their programming task. Results showed that with a clear assessment criteria students frustrations were reduced and anxiety elevated [30].

From this review we conclude that **there is a relationship between learning to program and anxiety**. When learning how to program a multitude of factors can contribute to feelings of anxiety in students such as: 1) receiving errors , 2) task complexity and 3) collaborative learning. Receiving any sort of error for the program just written can be a source of displeasure which could lead to anxiety. Not only is programming a source of anxiety - system response time and the program task has an effect on state anxiety.

5.2 Is there a relationship between mathematical anxiety and learning to program?

Of the 24 studies selected for this study, seven studies informed this research question. The studies are Owolabi et al. [32], Suraweera [38], Fone [15], Mills [28], Macher et al.[25], Vitasari et al.[41] and Baloglu et al.[2]

Owolabi et al. investigated the relationship between mathematical and programming anxiety by surveying students studying both computer science and mathematics [32]. They found a positive correlation between Mathematical anxiety and Computer anxiety (r=0.272). In addition, they found a strong correlation between mathematical anxiety and computer programming achievement ($\alpha = 0.01$, r = 0.450) [32].

Suraweera investigated Discrete Mathematics being taught by the Mathematics department in his institution. He noted that students were not understanding the material and subsequently could not apply the material in CS. He designed a framework to enhance the teaching and learning of Discrete Mathematics. This included Discrete Mathematics being taught by the CS department. After putting this framework into practice, students reported feeling more confident in their ability and less anxious [38].

Fone argued for reducing mathematical overheads to reduce unnecessary mathematical and programming anxiety [15]. The concept of neural networks is one that is rooted in Mathematics. Fone used Microsoft Excel to demonstrate to a class of 21 students, the operation of neural networks. Following this demonstration, the student's ability to program a neural network improved and reduced their reported programming and mathematical anxiety [15].

Macher et al. were interested in how self-efficacy and different learning strategies can influence mathematical learning [25]. As part of this study, 147 students participated (112 females, 35 males). Questionnaires on mathematics and trait anxiety, deep-level strategies, self-concept and interest in mathematics were administered. An interesting finding was that students with higher levels trait anxiety experienced higher levels of mathematical anxiety ($\beta = 0.42$). Along with mathematical anxiety, it was found that mathematical self-concept and an interest in mathematics are both negatively related to mathematical anxiety ($\beta = -0.22, \beta = -0.31$ respectively)[25].

Macher et al. also investigated the role of deep-level learning strategies (for example actively seeking to understand the material, interacting vigorously with the material etc.) on mathematical anxiety [25]. Statistical analysis showed that there is a relationship between the frequency at which students' applied this type of deep level learning and levels of anxiety ($\beta = 0.17$).

Mills noted that students were writing programs and not following a particular algorithm. This led to students not knowing if the program that they wrote would compile. In this his Mills outlined a mathematical technique that demonstrates how to know you have written a program correctly [28]. He discusses how if the students follow a systematic approach to writing a program it can aid in reducing feelings of anxiety.

Vitasari et al. and Baloglu et al. investigated the role that mathematical anxiety has on academic success. Vitasari et al. conducted a study with 770 students [41]. The aim of the study was to investigate the psychological barriers that students encounter when they are performing a mathematics task. They found that Mathematics is perceived as a difficult subject (t=72.414, p=0.000). Baloglu et al. conducted a study on 759 third level students to investigate the differences which exist in Mathematics anxiety. After administrating a survey, Baloglu et al. found that third level students doing basic maths such as multiplication and division still induced anxiety in students[2].

From this review we can conclude that **there is a relationship between mathematical anxiety and learning to program**. Many of the concepts that are taught in Computer Science have a basis in Mathematics. When students receive assignments that have a strong basis in Mathematics, they can find it hard to draw the link between what is being asked and how to complete the assignment.

Having a certain level of Mathematics in Computer Science is necessary. As part of introductory CS courses, Discrete Mathematics is often taught as a part of the course, something that is generally delivered by the Mathematics Departments. This practice has led to unsatisfactory results however. These poor results cause students to become more anxious in their ability.

5.3 Does computer usage cause anxiety when learning to program?

Four papers were found that were relevant to this question: Doyle et al. [12], Chua et al. [7], Maurer [26] and Todman et al. [40]. Many of the topics taught in CS involve using a computer. Even the basic interaction with a computer, using a virtual learning environment for example, may be enough to make a student anxious. Computer anxiety is defined as the "negative emotions and cognitions evoked in actual or imaginary interaction with computer based technology" [7].

A study by Doyle et al. investigated computer anxiety felt by CS students and how that anxiety is directly related to self-efficacy and computer experience [12]. Computer experience can include computer courses, computer training, computer gaming experience etc [12]. The study involved 163 participants (32 female, 131 male) across 4 different years in University. Students were asked to fill out a questionnaire and interestingly a strong inter-dependence between computer anxiety, self-efficacy and computer experience was found [12]. In addition, they found that final year Computer Science students are still anxious when it comes to completing a computer task.

Chua et al. conducted a review of 10 studies on the correlates of computer anxiety [7]. The review reports on the relationship between computer anxiety, age, gender and computer experience. They found that computer anxiety and computer experience are inversely related.

Maurer conducted a literature review on computer anxiety and its correlates [26]. The review consists of 38 studies. In the review Maurer discusses different correlates such as experience, gender, age, academic major, etc. Maurer reports that computer experience is a correlate of computer anxiety but still requires further research.

In the above reviews, gender was investigated as a correlate. While gender is not considered a strong correlate, Todman et al. has suggested that perhaps biological gender is not a variable in computer anxiety but psychological gender is. When examining psychological gender, each person would identify with one category: masculine, feminine, androgynous or indifferent. A study with 138 CS students was conducted. It was found that students who have a more feminine-identity experience a greater sense of computer anxiety. Overall, the gender factor on computer anxiety is inconclusive [40].

From this review we can conclude that **computer usage can cause anxiety when learning to program**. The relationship identified between computer experience and anxiety appears to be the strongest. Results from studies show that computer anxiety can be reduced through computer experience but it depends on the type of experience. Intuitively one might expect that the more experience you have in CS the less anxious you should be. However, final year CS students are still anxious when it comes to completing a computer task.

5.4 Does test anxiety affect programming and more broadly Computer Science students?

Three studies were found to inform this research question: Deloatch et al. [10], DeRaadt [9] and Kavakci et al. [22].

Test anxiety is an unpleasant state associated with the feeling of tension and apprehension, worrisome thoughts and the activation of the autonomic nervous system when an individual faces evaluative achievement demanding situations [22].

Deloatch et al. investigated how exam modality relates to students perceptions of test anxiety and performance during programming exams [10]. A survey was administered to measure student perception of test anxiety of paper based exams and online exams. Three hundred and ninety one students participated in this survey. After analysing the results, 22% of students (n=61, $\bar{x}$=4.26, SD=1.51) perceived high test anxiety for paper based exams while 23% of students (n=64, $\bar{x}$=4.15, SD=1.67) experience high test anxiety for online exams.

De Raadt proposed a method of allowing students to create cheat sheets for exams. Eighty nine students took part in the experiment. While exam marks did improve marginally, each student that created a cheat sheet reported that their levels of test anxiety reduced prior to and during the exam [9].

Fenwick et al. trialled a novel method of exam revision which consists of a 24 second technical description and a clear summary that anyone could understand in 7 seconds called a 24/7 lecture [14]. A total of 100 participants took part in the experiment. In order for the student to succeed in creating the 24/7 lecture the student had to completely understand the topic. Responses from the students showed that not only were they more confident in the concept they discussed but they felt less anxious[14].

Kavakci et al. investigated the variables that are related to students planning to take University entrance exams [22]. The aim was to identify the predictors of test anxiety. A total of 436 students participated in the study. They found that 48% of students experienced test anxiety.

From this review we can conclude that **test anxiety does affect programming**. In recent years, computer based exams have become more prevalent. The impact of the modality used to assess students in programming is currently unknown and future research is required.

6. DISCUSSION

In this review we have documented evidence on the relationship between programming anxiety and programming-task complexity and how this relates to programming skills. A significant relationship between perceived task complexity and self-perceived anxiety levels has been identified. In addition a longitudinal study which investigated the variance of anxiety amongst undergraduate computing students was described. It was found that students have low levels of self-belief when conducting programming tasks. This is compounded by evidence that students are leaving university as anxious programmers and going into industry lacking confidence in their own ability [3].

While it is known that programming is difficult, with the introduction of group work, students appear to be anxious about their work being examined by their peers [34].

Mathematical anxiety was also examined due to the close relationship of programming and Mathematics. Students are anxious about the teaching methods used [32]. In addition to the teaching methods, self-efficacy was identified as a key factor when learning Mathematics and consequently reducing Mathematical anxiety. In addition the learning strategies employed by students influences levels of mathematical anxiety.

Given the high availability of technological devices one would assume that students who choose CS as a degree choice would not fear interacting with a computer. However, even after a four year degree, students still feel anxious when working on a computer.

Testing and assessment can induce anxiety. Anxiety in assessment is inevitable, however, educators are now beginning to change the modality of how programming is being assessed. One study has observed that online assessment marginally reduces anxiety in students when programming is being assessed however the differences between paper based assessment and online assessment are still unclear [35].

The findings here can inform the teaching and learning of programming and help us to be mindful of the role of anxiety and its implications in learning [36].

7. TEACHING IMPLICATIONS

While there are only 24 studies to draw on, the following implications are noted for practitioners.

7.1 Is there a relationship between learning to program (language, syntax, compilation, etc.) and anxiety?

Student anxiety affects learning and programming is no exception.

1. As educators we need to consider how we can reduce the perceived complexity of assignments to reduce anxiety.

2. When learning to program, a responsive machine is needed to reduce or at least not increase anxiety.

3. If group work and peer assessment are used, it is important for educators to provide clear and transparent objectives and marking schemes for students to follow.

7.2 Is there a relationship between mathematical anxiety and learning to program?

Mathematical anxiety does effect levels of anxiety when learning to program. Of note:

1. CS has a strong foundation in mathematics. As such, perhaps we need to be mindful of mathematical anxiety. Techniques to alleviate this could include using real world examples of abstract concepts.

2. Discrete Mathematics can be problematic. As discussed earlier potentially teaching the subject within a CS department may reduce stress and anxiety.

7.3 Does computer usage cause anxiety when learning to program?

To reduce computer anxiety, students should get as much hands on computer experience throughout the course with appropriate support.

7.4 Does test anxiety affect programming and more broadly Computer Science students?

It appears that the modality tests (paper vs online) relates to anxiety. Being aware of this can reduce stress and anxiety for students. Novel study techniques such as the 24/7 lectures and development of cheat sheets have been shown to reduce anxiety and improve the understanding of concepts.

8. CONCLUDING REMARKS

This paper makes several valuable contributions. First this meta-review provides insight and promotes awareness of the anxiety of our learners. This can be used to improve our teaching and learning methods and assessment decisions. Second, given the concerns of the mental health of our students, we have identified how little focus has been given to such an important area. Third, this paper provides an example of how to perform a systematic review. We hope this will encourage researchers within our Computer Science education community to perform rigorous, reproducible and explicit reviews in the future.

Clearly there is a need for more research to be carried out in this area. In particular, two types of studies would be very valuable: further empirical studies that build upon the findings identified in this review are needed as well as experimental studies that involve real time physiological measurement of anxiety using sensor technology. For example, measuring heart rate or electro-dermal activity during ecologically valid programming tasks. Doing this during the completion of a task may give us an understanding of the specific source of anxiety or stress. This is particularly important now as wearable technology has become widely available and is potentially an untapped resource in teaching and learning.

9. ACKNOWLEDGEMENTS

This work was funded by the Irish Research Council Postgraduate Scholarship 2015 GOIPG/2015/1671.

10. REFERENCES

[1] Y. Amichai-Hamburger and E. Ben-Artzi. Loneliness and Internet use. *Computers in Human Behavior*, 19(1):71–80, January 2003.

[2] M. Baloglu and R. Kocak. A multivariate investigation of the differences in mathematics anxiety. *Personality and Individual Differences*, 40(7):1325–1335, 2006.

[3] A. Begel and B. Simon. Novice software developers, all over again. In *Proceeding of the fourth international workshop on Computing education research*, volume 1, pages 3–14, New York, New York, USA, 2008. ACM Press.

[4] J. Bennedsen and M. E. Caspersen. Failure rates in introductory programming. *ACM SIGCSE Bulletin*, 39(2):32, June 2007.

[5] S. Bergin. *Statistical and Machine Learning Models to Predict Programming Performance*. PhD thesis, 2006.

[6] S. E. Chang. Computer anxiety and perception of task complexity in learning programming-related skills. *Computers in Human Behavior*, 21(5):713–728, September 2005.

[7] S. L. Chua, D.-T. Chen, and A. F. Wong. Computer anxiety and its correlates: a meta-analysis. *Computers in Human Behavior*, 15(5):609–623, September 1999.

[8] C. Connolly, E. Murphy, and S. Moore. Programming Anxiety Amongst Computing Students - A Key in the Retention Debate? *IEEE Transactions on Education*, 52(1):52–56, February 2009.

[9] M. de Raadt. Student created cheat-sheets in examinations: Impact on student outcomes. In *Proceedings of the Fourteenth Australasian Computing Education Conference*, ACE '12, pages 71–76, Darlinghurst, Australia, Australia, 2012. Australian Computer Society, Inc.

[10] R. Deloatch, B. P. Bailey, and A. Kirlik. Measuring Effects of Modality on Perceived Test Anxiety for Computer Programming Exams. *SIGCSE '16 Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, pages 291–296, 2016.

[11] B. A. Dooley and A. Fitzgerald. *My world survey: National study of youth mental health in Ireland.* Headstrong and UCD School of Psychology, 2012.

[12] E. Doyle, I. Stamouli, and M. Huggard. Computer anxiety, self-efficacy, computer experience: an investigation throughout a computer science degree. *Proceedings Frontiers in Education 35th Annual Conference*, pages 3–7, 2005.

[13] K. Falkner, N. J. Falkner, and R. Vivian. Collaborative learning and anxiety. In *Proceeding of the 44th ACM technical symposium on Computer science education*, page 227, New York, New York, USA, 2013. ACM Press.

[14] J. B. Fenwick, C. Norris, A. R. Dalton, and W. Kreahling. 24/7 lectures as an exam review technique. In *Proceedings of the 41st ACM technical symposium on Computer science education*, page 455, New York, New York, USA, 2010. ACM Press.

[15] W. Fone. Using a familiar package to demonstrate a difficult concept. *ACM SIGCSE Bulletin*, 33(3):165–168, September 2001.

[16] C. Gerritsen, J. de Man, and J. van der Meij. Physiological and Subjective Response to Injustice: The Effects of Unjust Evaluations on Physiological Responses and Subjective Experiences. In *International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, volume 3, pages 179–182. IEEE, November 2013.

[17] M. Grigg and S. Saxena. Promoting mental health nursing research in low and middle income countries. *International Nursing Review*, 51(4):194–195, December 2004.

[18] J. L. Guynes. Impact of system response time on state anxiety. *Communications of the ACM*, 31(3):342–347, March 1988.

[19] J. Hamer, H. C. Purchase, P. Denny, and A. Luxton-Reilly. Quality of peer assessment in CS1. In *Proceedings of the fifth international workshop on Computing education research workshop*, page 27, New York, New York, USA, 2009. ACM Press.

[20] L. J. Heller, C. S. Skinner, A. J. Tomiyama, E. S. Epel, P. A. Hall, J. Allan, L. LaCaille, A. K. Randall, G. Bodenmann, C. W. P. Li-Tsang, K. Sinclair, J. Creek, L. C. Baumann, A. Karel, G. Andersson, R. Hanewinkel, M. Morgenstern, P. Puska, R. S. Bucks, J. Carroll, Y. Gidron, L. Rosenberg, A. M. Delamater, Y. Gidron, B. Spring, M. J. Coons, J. Duncan, A. Sularz, J. Deary, J. O. Prochaska, M. Spiers, E. W. Reid, R. Miller, C. Kirschbaum, C. Barrett, N. Rohleder, J. Wessel, L. Meneghini, E. R. Pulgaron, A. M. Delamater, S.-i. Suzuki, Y. Kunisato, and J. Denollet. *Trait Anxiety*, chapter Trait Anxiety, pages 1989–1989. Springer New York, New York, NY, 2013.

[21] P.-H. P. Hsieh, J. R. Sullivan, D. A. Sass, and N. S. Guerra. Undergraduate Engineering Students' Beliefs, Coping Strategies, and Academic Performance: An Evaluation of Theoretical Models. *The Journal of Experimental Education*, 80(2):196–218, January 2012.

[22] O. Kavakci, M. Semiz, A. Kartal, A. Dikici, and N. Kugu. Test anxiety prevalance and related variables in the students who are going to take the university entrance examination. *Dusunen Adam: The Journal of Psychiatry and Neurological Sciences*, 27(4):301–307, December 2014.

[23] B. Kitchenham and S. Charters. Guidelines for performing systematic literature reviews in software engineering, 2007.

[24] M. Liston, D. Frawley, and V. Patterson. *A Study of Progression in Irish Higher Education*. 2016.

[25] D. Macher, M. Paechter, I. Papousek, and K. Ruggeri. Statistics anxiety, trait anxiety, learning behavior, and academic performance. *European Journal of Psychology of Education*, 27(4):483–498, 2012.

[26] M. M. Maurer. Computer anxiety correlates and what they tell us: A literature review. *Computers in Human Behavior*, 10(3):369–376, September 1994.

[27] U. Melin and S. Cronholm. Project oriented student work. *ACM SIGCSE Bulletin*, 36(3):87, September 2004.

[28] H. D. Mills. How to write correct programs and know it. *ACM SIGPLAN Notices*, 10(6):363–370, June 1975.

[29] Y. Mino, T. Tsuda, A. Babazono, H. Aoyama, S. Inoue, H. Sato, and H. Ohara. Depressive States in Workers Using Computers. *Environmental Research*, 63(1):54–59, October 1993.

[30] G. Ngai, W. W. Lau, S. C. Chan, and H.-v. Leong. On the implementation of self-assessment in an introductory programming course. *ACM SIGCSE Bulletin*, 41(4):85, January 2010.

[31] K. Nolan, A. Mooney, and S. Bergin. Facilitating student learning in Computer Science : large class sizes and interventions. *International Confernce on Engaging Pedagogy*, 2015.

[32] J. Owolabi, P. Olanipekun, and J. Iwerima. Mathematics Ability and Anxiety, Computer and Programming Anxieties, Age and Gender as Determinants of Achievement in Basic Programming. *GSTF Journal on Computing (JoC)*, 3(4):47, April 2014.

[33] K. Quille, S. Bergin, and A. Mooney. Press#, a web-based educational system to predict programming performance. *International Journal of Computer Science and Software Engineering (IJCSSE)*, 4(7):178–189, 2015. This work is licensed under a Creative Commons Attribution 3.0 Unported License.

[34] R. Rickenberg and B. Reeves. The effects of animated characters on anxiety, task performance, and evaluations of user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, volume 2, pages 49–56, New York, New York, USA, 2000. ACM Press.

[35] M. J. Scott and G. Ghinea. Measuring enrichment. In *Proceedings of the tenth annual conference on International computing education research*, pages 123–130, New York, New York, USA, 2014. ACM Press.

[36] D. A. Sousa. *Mind, brain, & education: Neuroscience implications for the classroom.* Solution Tree Press, 2014.

[37] A. Stevenson. *Oxford Dictionary of English.* Oxford University Press.

[38] F. Suraweera. Enhancing the quality of learning and understanding of first-year mathematics for computer science related majors. *ACM SIGCSE Bulletin*, 34(4):117, December 2002.

[39] S. Thomée. *ICT use and mental health in young adults.* 2012.

[40] J. Todman and K. Day. Computer anxiety: the role of psychological gender. *Computers in Human Behavior*, 22(5):856–869, September 2006.

[41] P. Vitasari, T. Herawan, M. N. A. Wahab, A. Othman, and S. K. Sinnadurai. Exploring Mathematics Anxiety among Engineering students. *Procedia - Social and Behavioral Sciences*, 8(5):482–489, 2010.

[42] P. Vitasari, M. N. A. Wahab, A. Othman, T. Herawan, and S. K. Sinnadurai. The relationship between study anxiety and academic performance among engineering students. *Procedia - Social and Behavioral Sciences*, 8(5):490–497, 2010.

[43] C. Watson and F. W. Li. Failure rates in introductory programming revisited. In *Proceedings of the 2014 conference on Innovation & technology in computer science education - ITiCSE '14*, pages 39–44, New York, New York, USA, 2014. ACM Press.