Factors affecting programming anxiety amongst students education essay

The art or skill of programming is seen to play a very fundamental role in the third year Information Systems (IS), Systems Development Project that students need to pass in order to complete their undergraduate degree. Students are required to build a complete and functional system using the skills they have learnt and acquired from their first, second and third year computer-programming courses. This paper aims to evaluate the learning and educational factors affecting programming anxiety amongst students in IS group projects, with a particular focus on IS students in their final year of undergraduate study. The paper also seeks to compare and contrast findings from a survey conducted on 15 previous IS undergraduate students, with various literature reviews of current workings focusing on the difficulties of teaching and learning programming. Various themes have emerged from both literature and the research questionnaires, attempting to explain some of the main causes of computer programming anxiety amongst students in computing degrees. The most recurrent of these themes appeared primarily within the learning and educational (teaching) bracket. Key factors such as the teaching and learning environments (teaching methods and learning styles) of the students and educators, the students own personal learning environment and finally the pressured environment (external and internal) are analysed in great depth and compared and contrasted against our own personal experiences and the surveyed results from the questionnaire. Conclusions and recommendations are also drawn up that may be useful to both educators and students interested in exploring this area.

Keywords: programming teaching and learning, anxiety, learning styles

Introduction

Computer programming is a very necessary skill in the field of Computer Science (CS) and IS software development; however it is not always easily attainable, and as a result, often induces anxiety in students learning it (Gomes & Mendes, 2007). Researchers have over the years, extensively investigated the theme of programming anxiety, and have offered insight into the difficulties experienced by students in learning to program.

The low retention rate of CS and IS majors has been for many a cause for great concern (Connolly, Murphy, & Moore, 2009; Gomes & Mendes, 2007b). Despite the fact that an extensive amount of research has been done in and around the field of computer programming, it still appears to be a disconcerting problem (Mead et al., 2006).

Anxiety occurs as a result of mistaken judgment of a situation. Therefore, students suffering from (programming) anxiety often misjudge their abilities to learn programming, thus causing a blockage in their mental schema, which is responsible for allowing the ability to construct programs (Connolly et al., 2009).

Apart from the psychological aspect of the problem, much of the research conducted in literature focuses on establishing new teaching tools and methodologies to tackle the issue of programming anxiety (Rogerson, 2009).

The goal of this paper will be firstly to review and evaluate educational/learning factors influencing programming anxiety amongst IS students and secondly to compare and contrast the findings in literature with

personal experiences and surveyed views arising from the third year IS group project.

To accomplish this, the paper has been divided into three main themes. The first being the educational contributors of programming anxiety, with particular focus placed on the teaching tools, learning styles and teaching methodologies. Secondly a look at the students learning environment which focuses on issues such as the students previous exposure to the subject (programming), students learning " GAP" (between high school and university/ higher education) and the preparedness of the student. The final key point in this paper looks at pressures, both internal and external, and their effect on a students programming anxiety.

Some conclusions and recommendations, exploring possible ways of tackling this dilemma, are also given.

This paper seeks to offer insight on the topic of computer programming anxiety and would be of beneficial interest to individuals, particularly educators, who seeks to enhance their understanding of the difficulties associated with programming anxiety amongst IS students.

Learning / Educational Factors

Teaching Method

Students use different learning styles to learn or adapt to a particular subject. These learning styles vary between subjects and can therefore be greatly influenced by the teaching methods used by the educator (Jenkins, 2002). This section of the article examines teaching methods evaluated in

various literature reviews. A look at these teaching methods allows the opportunity to evaluate and observe varied perspectives and opinions from different authors on the teaching tools and learning theories associated with computer programming.

Teaching Tools and Learning Styles

To overcome the difficulties and obstacles associated with programming, educators need to evaluate their methods and tools of teaching programming (Rogerson, 2009).

With computer programming being as complicated as it is, there has been a substantial amount of research done on various teaching methodologies. One particular study by Gomes and Mendes (2007a) focuses on certain aspects educators need to explore in order to enhance their students problem solving abilities. Educators, as the author's state, need to focus on the individual student, placing particular emphasis on the student's knowledge level and their learning style.

Learning styles can be separated in to two main categories namely " deep" learning and " surface" learning (Marton & Saljo, 1979). A student using the deep learning approach tends to focus more on understanding a topic in depth, whereas the student making use of the shallow or surface approach attempts to focus on the facts of the topic and reproduce exactly what they have learned, this is often otherwise referred to as memorising. Therefore, students making use of the deep learning approach are relatively better of than those using the surface approach (Simon et al., 2006).

It is very important to note that while this classification may work well for subjects that are a body of knowledge e. g History, which requires learning and analysing particular dates, it does not however work for programming, as it (programming) is a skill rather than a body of knowledge. Programming as a learnt skill requires a simultaneous approach of both learning styles.

Another very important and powerful teaching tool highlighted in the same study, involves the use of games as an aid to help develop students' cognitive problem solving skills. Games have the ability to " increase students' motivation and engagement in learning to program" (Eagle & Barnes, 2008; Eagle & Barnes, 2009).

Although not every teaching tool is examined and evaluated in this paper, it is still apparent that there exist a conglomerate of teaching tools, which are structured very differently, but all aim to have the same result – improve programming skills. It is therefore the educator's responsibility to ensure that students adopt the most appropriate learning method that not only strengthens there understanding of the subject but also motivates them to succeed (Jenkins, 2001).

Learning Theories

Learning to program requires the adoption of a new skill, and this, as many educators have noted, is one of the major reasons why students experience difficulty in this field (Lister, Simon, Thompson, Whalley, & Prasad, 2006).

It is as much an intricacy to learn a new programming skill as it is to teach it (Allison, Orton, & Powell, 2002). This phenomenon arises subsequently due to the fact that " traditional teaching methods do not adapt well to the

domains of coding and problem solving, as it[programming] is a skill best learned through experience" (Travnor & Gibson, 2004).

Various literature is written on the subject of learning theories and their applicability in terms of (student) learning. It stands that the most significant of these theories is that of Dreyfus and Dreyfus (2001), on the five stages of learning. The focus for this paper however, will be on Cockburn's (2002) three stages of learning.

According to Cockburn, individuals go through three main stages of learning, namely, following, detaching, and fluency. In the following stage, individuals gradually try to learn and adapt to the new skill by following instruction. Once the individual is confident enough to carry out the skill without the aid of instructions, they detach and are now able to learn any new concepts or rules appropriate to skill. The final stage in Cockburn's learning theory is that of fluency, at this point an individual can be considered an expert, as they have fully grasped and become fluent in their particular skill.

The above theory can be closely linked to the skill of learning to program. Students should in theory go through each of these stages and become " expert coder" however, in a study by Thomas et al. (2004) it was revealed that even with these frameworks and tools, certain students, usually the weaker one's, are still unable to reach the fluency stage due to other underlying problems.

Student Learning Environment

Previous exposure

According to Hagan and Markham (2000), the lack of prior computer experience appears to be a significant disadvantage for many students, with the success factor increasing systematically in relation to the number of programming languages to which a student had been exposed to. Both Byrneand Lyons (2001) and Simon et al. (2006) concur with this finding.

Previous exposure to programming has an effect on how well a student performs at tertiary level (Howerton, 1988). For most students, the initial encounter with computer programming is often established for the first time at university, hence they lack any skills that would help them understand and form the needed mental schemas. This factor then escalates, causing students to develop uneasiness when learning to programme. The situation is further worsened by the repeated failure of the programmes, that they will be developing, to run positively as expected (Connolly et al., 2009). Research has provided proof that a strong association exists between previous programming background and success in programming courses at university level. The same study continues to attribute that such students have a success rate of 70% as compared to 52% with those that have not taken any programming course before (Morrison & Newman, 2001).

Results from our questionnaire indicate that students in IS groups from the previous year (2009) were of the following mix with regards to previous programming exposure (see table 1). 50% of the student, who completed the survey, did not have previous programming skills whatsoever when they

came to university, and 50% indicated that they had acquired the skills at high school.

On further analyses, it was discovered that the students who had prior experience, found programming to be a pleasant experience, and had little or no negative perception about it. On the other hand, some of the participants within the group with no prior experience, apart from venting out their frustration about programming, went on to say that, they did not find any interest in the course and only did it because it was required of them, ” I did It (programming as a course – I’m not really interested in it”

In comparing and contrasting the findings from the survey with those from literature, it is clear that there is an agreement between the two. Having a good programming background (especially from high school) is necessary for laying the foundation to the logical and syntactic thinking needed for solving programming problems.

In questionnaire

Have anxiety

Programming exposure

50%

35%

No prog. exposure

50%

65%

Table1: Impact of programming exposure on anxiety on IS groups

Learning Gaps

A student's decision to pursue a career in the Information Technology (IT) industry is often strongly influenced by their high (secondary) school education. It is here that students acquire the initial foundation of skills and knowledge that can be further developed by a higher institution of learning (e. g. University) (Brinda et al., 2008).

The problem, however, is the gap between what is taught in the secondary school and what is necessary for University. Many of the participants, who had prior programming experience, felt that the body of knowledge attained in secondary school was insufficient, and not very well taught.

Preparedness

Research has shown, that students' who continuously practise their programming skills, easily reduce their anxiety since they gradually attain confidence in their skills (Ramalingam et al., 2004). When these skill are not further improved upon, the student's level of anxiety will continue rising up to the point where even the mentioning of the word programming itself result in a state of uneasiness (Ramalingam et al., 2004).

From our survey, we discovered that just over 80% of the participants hardly ever put much effort into practising their programming skills, whereas another 20% indicated that they often invested more time into practising programming on their spare time, and this helped them to become better at it.

Pressures (Demand Factors)

Very few educators within the computing field would dispute that the skill of programming is not always as easy as it seems (Jenkins, 2002). Much of the literature written on the topic of programming focuses mainly on exploring and establishing new and better ways of teaching the subject. However, as Jenkins (2002) points out, " if educators hope to teach effectively, they must understand precisely what makes learning to program so difficult for so many students".

This section of the paper looks at the Pressures that play a role in influencing programming anxiety amongst students. According to Rogerson (2009), programming anxiety is affected by both internal and external factors. The evaluation of these factors allows the opportunity to better understand some of the pressure related barriers making programming more difficult for some students.

Internal Factors

In examining the internal pressures, focus is placed on factors that are directly attributed to the student; in other words, these are usually factors the student has control over and no other external forces such as an institution, an educator, or a peer can alter or change them (www. answers. yahoo. com, 2009).

The internal factors identified under this theme are [1] cognitive learning ability, [2] learner attitude, [3] self-efficacy, and [4] motivation. Although there exists many other factors, the following have been identified as being the most prominent within our research survey.

Cognitive learning Ability

Cognitive factors identified as possible predictors of programming anxiety include problem solving ability, abstract reasoning, ability to understand problems, logical ability and cognitive style (Gomes & Mendes, 2007a).

A study by Gomes and Mendes (2007b) explains that one of the major reasons why students find it difficult to learn programming is because many of them " lack generic problem solving skills". This subsequently hinders students from being able to create algorithms, as they are unable to logically solve and understand the problems.

In terms of the students' logical ability, literature has shown that there exists a general positive relationship between a student's mathematical problem solving ability and successful programming (Boyle, Carter, & Clark, 2002; Mcgettrick et al., 2005; Simon et al., 2006).

The results from the survey differ quite substantially however from what is presented in literature. Many of the participants in our study indicated that they had a good mathematical solving ability; however they still struggled with programming.

Attitude

The term " attitude" refers to the favourable or unfavourable feeling that students have towards the subject (Fesakis & Serafeim, 2009). A student's attitude often determines how well they able to handle difficulties experienced within the course (Wiedenbeck, 2005). From our surveyed, it was evident that a student with a positive attitude is more likely to overcome

their anxiety of programming. Our view is that a positive attitude often acts as a motivator in a difficult situation. In the study conducted by Rogerson (2009), participants expressed that having the right attitude was necessary in helping to overcome their difficulties with programming.

Self-Efficacy

Self-efficacy is the belief in one's ability to undertake a particular task (Rogerson, 2009; Connolly et al., 2009). Bandura (1994, p. 36) concurs this through the following statement, " perceived self-efficacy is concerned not with the number of skills that you have, but with what you believe you can do with what you have under a variety of circumstance". He further asserts that there exists a proportional relationship between a student's self-efficacy and their choice of activities. A student with high self-efficacy is more motivated to take on challenges, as they possess a stronger belief in themselves to succeed.

The findings from the survey showed that the majority of the participants were motivated and determined to overcome their difficulties with programming, " I really love it, just not as good as I should be"

Motivation

The computing degree is associated with many negative connotations, some of which may cause many potential students to shy away from the course as it may affect their social image (Jenkins, 2002). One very notable connotation is that computer programming is a very difficulty course (Gomes & Mendes, 2007c). This causes students to enter into the course with very negative perceptions that often lead to increased anxiety.

Therefore, it helps to motivate students and encourage them to endeavour even when things seem bleak.

“ Programming students are motivated to succeed; they do not fail on purpose (Jenkins, 2001).”

The results from our survey show that students often feel encouraged to practice and understand programming when they have a fierce motivational system, this includes, motivation from family members, friends, peers, teachers and tutors.

External Factors

In examining the external pressures, focus is placed on factors that are indirectly attributed to the student and the student has no control over. The external factors identified under this theme are [1] student support, which consists of support from lectures and tutors, and [2] teaching methodology.

Student Support (Lecturer/Tutor)

Student support according to Tait (2000) refers to the various services offered to individuals and student groups, which correspond to the learning resources or course materials that are uniform for all learners. In Tait’s (2000) paper, he identifies three primary functions of what student support should encompass: [1] cognitive: Supporting and developing learning for individual students, through the mediation of the standard and uniform elements of course materials and learning resources [2] Affective: The ability to provide the students with an environment which supports them, stirs up commitment within them and boosts their confidence levels [3] Systemic:

Implementing an administration process and a knowledge management system which proves to be transparent, effective and efficient as well as user (student) friendly.

The IS department offers student support in the form of lecturing, which takes the form of a instructor-learner environment; and tutoring, which involves more experienced students offering a more individual based teaching method; interactive workshops and hot seats with tutors are also offered.

Looking at the outcomes of our survey, students felt that the support services in place for third year IS students were sufficient, although some felt that the learning materials they received were not useful as they were not always sure how to use them. And would thus result in them simply “ cutting and pasting” the required code from their word documents instead of understanding the code. The students who felt this way were generally the ones who do not enjoy programming.

Teaching Methodology

In the so called ‘ post-industrial societies’ student learning and teaching continue being the most essential activities in the ever more intricate and dexterous educational systems (Hager, 2005, p. 633). When looking at the fields of Computer Science (CS) and Information Systems (IS), there has been a great deal of apprehension regarding the learning and teaching of computer programming languages, and the deteriorating rates associated with the above mentioned subjects (Robins, Rountree, & Rountree, 2003).

According to our research, more students felt that the methodology their lecturer used to teach programming had a huge impact on the understanding they gained from the various courses.

Where a few participants felt that the teaching methodology used by the lecturer was sufficient, a higher number of participants felt that their ability to grasp the fundamentals of programming was hampered by the teaching methodology used.

Outcomes of Anxiety

In the survey, we asked participants how they thought their programming anxiety hindered/affected them personally and or professionally. The results from the survey showed that a vast majority of the students felt that their inability to program affected their self-efficacy. Other internal factors such as motivation and attitude also become victim to the anxiety.

In a study done by Rogerson (2009) it was found that students who generally find it hard to program also lack internal motivation. The anxiety associated with programming often leads to other negative psychological and or emotional factors such as the feeling of alienation from one’ s self and peers the and the lack of confidence which may lead to them having a negative perception of programming.

Conclusion and Recommendations

This paper has presented literature that shows various themes attempting to explain some of the main causes of computer programming anxiety amongst

students in computing degrees. The most recurrent of these themes appeared primarily within the learning and educational (teaching) sections.

The study found that although students were exposed to various factors both internally and externally to their environment, the most influential of these factors, attributing to their programming anxiety leaned towards the student support systems in place, and the teaching methodologies used in the course. Other significant contributors included the students' personal confidence or self efficacy and the students' previous programming exposure.

The findings of this paper and its implications on the IS discipline as whole are very crucial. They leave more to be desired in the teaching/learning environment that students find themselves in while learning programming skills in IS. The evaluation of educational factors that affect students in IS groups agrees with most work that has previously been done on this subject, but the main issue is what should be done to implement these important findings; thereby helping to reduce or even avoid the level of anxiety that the IS students are experiencing.

Connolly et al. (2009: 55) states that there needs to be a shift of focus from trying to change the external elements of teaching techniques to creating environments that allow students to feel comfortable and thereby reducing their anxiety.