Association for Information Systems
AIS Electronic Library (AISeL)

SAIS 2004 Proceedings | Southern (SAIS)

3-1-2004

Personality as a Predictor of Student Success in Programming Principles

Amy B. Woszczynski
awoszczy@kennesaw.edu

Tracy C. Guthrie

Tai-Lung Chen

Sherri Shade

Follow this and additional works at: http://aisel.aisnet.org/sais2004

Recommended Citation

Woszczynski, Amy B.; Guthrie, Tracy C.; Chen, Tai-Lung; and Shade, Sherri, "Personality as a Predictor of Student Success in Programming Principles" (2004). *SAIS 2004 Proceedings*. 1.
http://aisel.aisnet.org/sais2004/1

This material is brought to you by the Southern (SAIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in SAIS 2004 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

PERSONALITY AS A PREDICTOR OF STUDENT SUCCESS IN PROGRAMMING PRINCIPLES I

Amy B. Woszczynski
Kennesaw State University
awoszczy@kennesaw.edu

Tracy C. Guthrie
Kennesaw State University
tguthri@bellsouth.net

Tai-Lung Chen
Kennesaw State University
txc6272@students.kennesaw.edu

Sherri Shade
Kennesaw State University
sshade@kennesaw.edu

Abstract

Large numbers of college students continue to fail to successfully complete programming principles courses. However, little research has addressed potential reasons for student failure. Many educators simply assume that high failure rates are acceptable – that computer programming is difficult and some students simply "don't get it." Some researchers (i.e., Bishop-Clark & Wheeler, 1994; Carland & Carland, 1990) have studied personality as a predictor of success in computer programming courses. However, with the exception of Woszczynski & Guthrie (2003), few studies have attempted to gather cognitive profiles (Krause, 2000) and match performance to profile type exhibited. Krause's work shows that students with identified profiles can apply certain study skills to improve the probability of success in the classroom, and Woszczynski & Guthrie (2003) extended this research to the programming classroom, identifying underperforming cognitive profile groups. This study identified the primary cognitive profile of 236 students in a programming principles course at a southeastern university and matched profile to final average in programming principles I. Overall, intuitive thinkers (NT) tended to perform better in programming principles I than sensor feelers (SF). We found no other differences in performance between other paired profiles. We recommend a number of interventions to reach underperforming groups.

Keywords: programming principles, cognitive profiles, personality, CS1

Introduction

Dr. Carl Jung (Campbell, 1971) completed some of the earliest work on cognitive profiles. His studies attempted to group personality profiles into structures or combinations of one selection from four pairs of possibilities. These early studies led to the use of Jung's profiles in the Myers-Briggs Type Inventory (Corns, 1998; Ring, 1998), which is sometimes used to study how people interact in society.

This paper extends Jung's work by applying personality research and in particular, cognitive profiles, to an analysis of a typical "gatekeeper" course in computer science education, programming principles I – often referred to as simply CS1. Across universities and curricula, CS1 has a notoriously low rate of success – defined here as earning an A, B, or C in the course. Personality research offers great potential for giving educators and researchers more information on why so many students fail to succeed in CS1. Krause (2000) demonstrated that students may learn in different ways based on their personality profile, and this research identifies potential interventions to reach consistently underperforming groups.

For this project, CS1 students took either a paper and pencil or online version of the Cognitive Profile Inventory (CPI) (Krause, 2000). Then we tracked student performance in the course by personality type to identify groups that might need different pedagogy and interventions to reach their full potential for the successful completion of CS1. Next, we recommended corresponding interventions to reach groups that performed poorly. We believe that appropriate instructor pedagogy and interventions can improve the success rate of intellectually capable and properly motivated students. The next section provides a brief overview of previous research into computer programming, personality, and cognitive profiles.

Theoretical Background

Difficulty in Computer Programming

Computer programming includes many aspects of learning as it requires the prospective programmer to analyze problems, implement solutions in a programming language, execute the solution in a computer operating system, track, follow, and debug code if necessary, and make enhancements to the program to further the effectiveness of the solution (CS1, 2002). Another way to look at the difficulty of learning how to write computer programs is to imagine receiving a document written in an unfamiliar foreign language with the assignment to read, process, and assign a solution for the stated problem.

Students in computer programming curriculums have traditionally struggled with one or many of the concepts required for success in this field. As evidence of this struggle, statistics show that failure and course withdrawal rates often exceed 50% or more for CS1. In a recent study, researchers reported on the pass, fail, and withdrawal figures for the CS1 classes where successful completion of the course requires a grade of C or better (Beise, Myers, VanBrackle, & Chevli-Saroq, 2003). They found that the overall probability of passing CS1 the first time was 40% across all majors, with an initial failure rate of 19.5%, and a withdrawal rate of 40.5%.

These same course statistics can be broken down even further into the individual majors that are required to participate in this course. Beise et al. found that computer science majors pass the beginning programming class 55.5% of the time, information systems majors pass the class 31% of the time, and others (mainly math majors) pass the class 33.5% of the time. Surprisingly, the failure rate in this course is also highest for computer science majors at 27.5%, with IS majors showing a 16% failure rate, and other majors at 15%. The dropout or withdrawal rate follows yet another trend with the lowest number, 17% coming from the CS majors, 53% from IS majors, and 51.5% from other academic fields.

On initial observation of these statistics, one could argue that regardless of the technical acumen that a student possesses, the failure rate in all fields is very high. Is this rate caused by the degree of difficulty involved in the course? Is it due to poor study habits on the student's part? Does this simply highlight that the learning of this subject is unique and one that students have limited exposure to until they reach the college level of instruction? Alternatively, do these findings and other similar studies point to the fact that all people learn differently, and respond to differing academic and environmental stimuli when learning in any specialized discipline? This study attempts to address some of these issues, after first discussing personality and cognitive profiles.

Personality

In Jung's work on personality profiles entitled *Psychological Types*, he sought to identify people as introverts or extroverts, where introverts relate more to interest in a subject, and extroverts focus more specifically on objects in their environment. That is, extroverts are more outgoing and sociable, while introverts may be withdrawn and shy. Later work has built upon Jung's pioneering efforts, resulting in a plethora of studies in organizational behavior and psychology (Campbell, 1971).

After identifying subjects as either introverts or extroverts, Jung then sought to break a person's psyche down into sections that are more detailed. The identified functions or thought processes are: thinking and feeling, which Jung argued were rational thought processes; and sensation and intuition that he felt were irrational thought processes.

Myers-Briggs took the initial findings of Jung and attempted to break them down even further into sixteen different personality types (Krause, 2000). One of the main differences between the Jung study and the Myers-Briggs study is that Myers-Briggs did not attempt to identify and group a person into one category, but felt that a person – while having a preference on each side of the four main categories – could cross the boundaries in a given situation. This corresponds to the widely regarded interactionist viewpoint, in which inherent personality characteristics and the situation in which a person is placed both play a role in how that person behaves (Bowers, 1973; Caspi, 1987; Funder & Dobroth, 1987; Magnusson, 1981).

Myers-Briggs has found high acceptance rates in many different areas, and is used quite regularly for businesses seeking to provide more in depth analysis of employment candidates (Harvey, Murry, & Markham, 1995).

Few would argue with the contention that some people exhibit artistic characteristics, some people have scientific tendencies, some people pursue more theoretical outlets, and others naturally succeed in mathematical endeavors. To a certain degree people's strengths cross these boundaries when they are skilled in multiple areas, but people always seem to have a weaker area of natural interest and innate ability. Most people find some areas extremely challenging with other areas seeming extremely easy. One method of addressing personality as it relates to improved study habits and greater success in the classroom is through the use of cognitive profiles, as described in the next section.

Cognitive Profiles

The book *How We Learn and Why We Don't* (Krause, 2000) attempts to identify four distinct learning types using a word identification test that also helps to determine the correct study skills for the established dominant cognitive profile. The test is simple in nature and involves nothing more than the participants selecting one word from sixty pairs of words and selecting which word better describes themselves. The numerical results from the test produce a map or graph demonstrating the degree to which the student's cognitive results fall into the different cognitive profiles.

The types of learners are discussed in two letter acronyms that include ST for Sensor Thinker, SF for Sensor Feeler, NT for iNtuitive Thinker, and NF for iNtuitive Feeler. The sensor attribute refers to people who easily learn or gain knowledge through one or more of the five primary senses – touch, taste, sight, sound, and smell, while intuitive learners use visual memory cues to trigger retention of information. Thinkers like to have concrete and solid evidence or information for decisions, while feelers tend to make decisions based on emotion, morality, or the perception that the choice will provide.

ST students prefer to study by themselves with little distraction in a clean work environment where all of the study resources are readily available. They learn through note taking and practice problems, which explains why these students perform well on tests that require retention of large amounts of material and where the answers are clear-cut. ST profiles have the ability to imitate tasks or patterns in a repeating fashion and will use this skill in learning new information. They will perform pre-test routines repeatedly until they are comfortable with the information or procedure. Through their goal of perfection, STs tend to perform well in occupations that require repetitive actions like airline pilots or surgeons.

Students in the SF profile group use structured thought processes and learn through repetition and breaking problems into steps or milestones. These types of learners attempt to relate the information to real life objects that they are familiar with and tend to perform well in occupations where they can assist other people, such as the education or medical fields. Just as SF profiles enjoy helping others, they also tend to prosper in the educational experience through studying and working in pairs or groups. These learning partnerships serve their purpose through rehearsing and talking through problems and questions.

NT students like to identify trends or patterns in the material using chapter summary information and visual aids such as charts and graphs to assist them in memorization. Often these learners are misdiagnosed as having learning difficulties, but in reality, they simply do not enjoy repetitive tasks such as memorization of material. With their abilities to apply pictures to learning, NT profiles often excel in fields related to architecture, engineering, and other occupations that require a person to see things before they exist physically. Based on the ability of NTs to see something before it exists physically, we believe that they will be able to "picture" how a program should be structured before it is actually written. This awareness of how the final product will look should help the NTs to plan and accurately develop a program and then code it properly.

NF students are the identified daydreamers or mind wanderers of the world. These types of students are adept at enhancing existing ideas or concepts, but sometimes have difficulty in explaining their ideas to others. NFs will also find difficulty when trying to place their ever-changing thoughts into productive ideas. They tend to respond to learning most effectively when allowed to build from nothing and when given the freedom to work through many ideas. Like NTs, NFs need to study the summarized and visual interpretations of data before attempting to gain more in depth knowledge of a subject. This type of learner will gravitate towards more artistic types of endeavors such as design, music, art, or language instruction.

Each profile learns, processes, and retains information using different skills and innate abilities. In the past, cognitive profiles have demonstrated the ability to determine how students will comprehend information in the most efficient and beneficial way. Because each profile is different in varying degrees from the other identified profiles, and because some profiles are naturally more adept at technical acumen, we predict:

Hypothesis 1: Students with diverse dominant cognitive profiles will perform differently in CS1.

Feeling profiles tend to react to problem solving based on their feelings, emotions, or through the use of sensory perception. On the other hand, thinking profiles tend to gather and digest all of the pertinent information before seeking a relevant solution. Success in CS1 generally requires the ability to collect and interpret information related to a problem, followed by the application of this information to a corrective action. Because thinking profiles are more inclined to process information in this manner, we predict:

Hypothesis 2: Students with Thinker (NT & ST) profiles will have higher success rates in CS1 than Feeler (NF & SF) profiles.

Methodology

This study determined the dominant cognitive profile of programming principles students at a southeastern regional university. Participants received either: 1) a free online cognitive profile analysis and feedback on the identified study skills that have proven effective for their dominant profile type in past cognitive analyses or 2) a free copy of the book *How We Learn and Why We Don't*, by Lois Krause (2002) and accompanying survey. Both surveys were identical. We then tracked these students through the assignment of final grades for CS1 to determine if success rates increased with student awareness of cognitive profile and use of recommended study tools.

Participants

A sample of 236 students completed the cognitive profile exercise. Of those, 70 of the respondents were female and 166 male. The sample included a variety of majors, with computer science and information systems comprising almost 75% of the respondents.

Results

We completed an analysis of variance (ANOVA) to test differences between groups. Hypothesis 1 was supported: Students with diverse dominant cognitive profiles performed differently in CS1 ($F=2.83$ and probability $> F = 0.0394$). Students with NT profiles achieved the highest overall final averages, with a mean grade of 2.92 on a 4.00 scale, where 4.00 is equivalent to a grade of A, 3.00 is equivalent to a grade of B, and so forth. Those with NF profiles achieved average grades of 2.64, while ST profiles averaged 2.60. SF profiles scored lowest, with an average grade of 2.11 out of 4.00.

To test Hypothesis 2: Students with Thinker (NT & ST) profiles will have higher success rates in CS1 than Feeler (NF & SF) profiles, we used several tests of differences between means. Tukey's Studentized Range test, t-tests, Duncan's multiple range test, Scheffe's test, and Student Newman-Keuls tests all showed significant differences between NT and SF cognitive profiles at the 0.05 level. That is, NT profiles scored significantly higher than SF profiles. None of the tests revealed any differences between other profile pairs. Therefore, Hypothesis 2 was partially supported. Intuitive Thinkers (NT) did score significantly higher as a group than Sensor Feelers (SF). However, we had also predicted that NTs would score significantly higher than Intuitive Feelers (NF), and that result was not supported by any of the pairwise comparison tests. Further, Sensor Thinkers (ST) did not score significantly higher than either SFs or NFs, as we had predicted.

Discussion

Potential Interventions

Our results seem to indicate that with the exception of the SF cognitive profile, all other profiles exhibit similarly positive outcomes in CS1. Therefore, the focus of our efforts should concentrate on bringing the SF profiles up to the level of the other profile groups. As mentioned previously, SF students tend to learn best through repetition and by breaking problems into steps or milestones. Therefore, these students might benefit from starting with very small steps in constructing a program. After mastering one concept or one sequence of instructions, they can add another piece. These learners should respond most favorably to a learning environment where they can receive credit for pieces of programs or for concepts learned. The movement to object-oriented programming (OOP) could provide SF students with the ability to develop these smaller modules of code as building blocks to solve larger or more complex problems. OOP could appeal to the SF profile and provide a more enriching CS1 experience for the learner who prefers to build and test solutions to problems in smaller steps. Moreover, the move to Web based and visual programming environments could also appeal to the SF learner as these alternatives build upon appearance that is artistic in nature, use small programming modules to meet specific needs, and have dynamic presentations to serve multiple users.

Since these types of learners attempt to relate the information to real life objects that they are familiar with, the use of more real-world problems in CS1 should also benefit them. Since real-world problems are typically of interest to diverse types of students and learning profiles, this step should also make the course more interesting and relevant to a wide variety of students.

In addition, since SF profiles tend to perform well when studying and working in pairs or groups, we see great potential for the use of innovative ideas in the classroom, such as paired programming. Since many programmers will find themselves working in teams when they enter the workforce, we contend that paired programming should also benefit all types of learners. Further, we should encourage the SF profile to find partners and study in groups to maximize the learning experience. Pairing SFs with other SFs may provide an enhanced outcome for multiple students. The SF learner might find it advantageous to attend outside laboratory environments and/or smaller, recitation-like reviews where they can receive additional practice – that is, repetition – in programming concepts. These environments give the SF learner the ability to ask multiple questions in a less-threatening environment while also giving multiple opportunities to practice the concepts covered in the classroom.

Finally, simply offering all students the opportunity to learn about their personality may help the students study and perform more effectively. Many universities offer free or reduced-price personality testing and career counseling, and students could be encouraged to use these services. Integrating an out of class on-line exercise that provides students with the opportunity to learn about themselves and how best to study to achieve success is also an option. None of these options should detract from the performance of the higher-achieving groups, and each of the options mentioned has minimal or no costs. Indeed, by offering a more welcoming environment to a diverse group of students, information systems and computer science students should benefit through the advent of new ideas and new types of individuals into the classroom and ultimately, into the working environment.

Limitations

Although we had a large sample size of 236 students, we had small numbers of the SF profile, with only 36 students, or about 15% of the sample. NT profiles represented the largest percentage of students, with over 30% of the sample. These results were not unexpected, however, since information systems and computer science students, who made up a large portion of our sample, often fall into the NT profile. Further, almost 75% of our respondents were either computer science or information systems majors. Future studies could sample a more diverse population to see if the results hold true with varying numbers of different profile types and across a variety of majors.

Further, our study also used self-reports to gather data on respondents. Since multiple studies have noted the inherent limitations of self-reports (Woszczynski & Whitman, 2004), we recommend gathering data using alternative formats to reduce bias caused by using a common method. Moreover, using longitudinal studies to track performance over time throughout the college career would help to overcome limitations of self-reports.

Conclusion

Our study does not attempt to find a way to reach all students who struggle with programming principles. We fully agree that some students are not meant to major in information-technology related fields. However, if we can modify the pedagogy and curriculum delivery mechanism for programming principles so that students who have the desire and intellectual capability to succeed are able to succeed, then we will increase the diversity in the field over time. By integrating some of the interventions recommended above, we believe that educators will be able to reach that portion of students who are fully motivated and capable of succeeding in programming principles, but who struggle with the method in which programming principles is presented. Will there still be a high percentage of students who fail to successfully complete programming principles? We believe the answer is yes. But will we reach a more diverse group and one that can further enrich the information technology field? We believe the answer to that question is also yes, and we hope that educators will implement, test and report upon some of the interventions that we have recommended.

References

Beise, C., Myers, M., VanBrackle, L, & Chevli-Saroq, N. (2003, forthcoming). An examination of age, race, and sex as predictors of success in the first programming course. *Journal of Informatics Education and Research*.

Bishop-Clark, C., & Wheeler, D. (Spring 1994). The Myers-Briggs personality type and its relationship to computer programming. *Journal of Research on Computing in Education, 26*(3), 358-370.

Bowers, K.S. (1973). Situationism in psychology: An analysis and critique. *Psychological Review, 80*, 307-336.

Campbell, J. (1971). Psychological types. In R.F.C. Hull (Ed.), *The Portable JUNG (pp. 178 – 269).* New York: Viking Press.

Carland, J. A., & Carland, J. W. (1990). Cognitive styles and the education of computer information systems students. *Journal of Research on Computing in Education, 23*(1), 114-126.

Caspi, A. (1987). Personality in the life course. *Journal of Personality and Social Psychology, 53*, 1203-1213.

Corns, S. (1998, December 8). Workshop shows what makes people tick. *Arizona Daily Wildcat*. Retrieved on October 20, 2002 from http://wildcat.arizona.edu/papers/92/73/13_1_m.html

CS1 – Programming Principles I Fall 2002 – Syllabus (n.d.). Retrieved on October 15, 2002 from http://science.kennesaw.edu/~mahmed/cs2301syl.html

Felder, R. (1988). Learning and teaching styles in engineering education. Retrieved October 21, 2002 from http://www.ncsu.edu/felder-public/Papers/LS-1988.pdf

Funder, D., & Dobroth, J. (1987). Differences between traits: Properties associated with interjudge agreement. *Journal of Personality and Social Psychology, 52*, 409-418.

Harvey, R., Murry, W., & Markham, S. (May 1995). A "Big Five" scoring system for the Myers-Briggs Type Indicator. Retrieved on November 10, 2002 from http://harvey.psyc.vt.edu/Documents/Personality/bigfive.doc

Krause, L. (2000). *How we learn and why we don't.* Cincinnati, OH. Thomson Learning

Magnusson, D. (Ed.) (1981). *Toward a psychology of situations: An interactional perspective*. Hillsdale, NJ: Erlbaum.

Ring, B. (1998). Myers-Briggs Type Indicator: A research report. *Personality plus*. Retrieved October 19, 2002 from http://members.tripod.com/~PersonalityInstitute/Myers-BriggsTypeIndicator.htm

Ross, K. (1997). Carl Gustav Jung (1875 –1961). Retrieved on September 23, 2002 from http://www.friesian.com/jung.htm

Ross, K. (n.d.). Psychological types (after C.G. Jung & the Briggs – Myers Typology). Retrieved September 23, 2002 from http://www.friesian.com/types.htm

Turns, J. & Wagnert, T. (August 2001). Computer science education research on programming: What we know and how we know it. PETTT Technical Report. Retrieved on September 30, 2002 from http://depts.washington.edu/pettt/papers/reviews/lori.pdf

Webster, R. (2001). Interfaces for E-learning: Cognitive styles and software agents for Web-based learning support. Retrieved September 15, 2002 from http://www.medfac.unimelb.edu.au/ascilite2001/pdf/papers/websterr.pdf

Webster, R. (2002). Learning styles and design: The use of ASSIST for reflection and assessment. Retrieved September 19, 2002 from http://www.ecu.edu.au/conferences/herdsa/papers/ref/pdf/Webster.pdf

Woszczynski, A., & Whitman, M. (2004). Chapter V. The problem of common method variance in IS research. In M. Whitman and A. Woszczynski (Eds.) *The handbook of information systems research.* Hershey, PA: Idea Group Publishing, 66-77.