Faculty and Industry Conceptions of Successful Computer Programmers

Gregory D. Sterling
Thomas M. Brinthaupt
Department of Psychology
Middle Tennessee State University
Murfreesboro, TN 37132 USA
gregsterling@hotmail.com tbrintha@mtsu.edu

ABSTRACT

Identifying success criteria for computer programmers can help improve training and development programs in academic and industrial settings. In the present research, we interviewed college faculty members and obtained a list of success criteria for both individual and group programming settings. Then, faculty and industry members rated the importance of these characteristics in each setting. The two settings showed both common and unique success criteria. Shared criteria included being creative and conscientious and enjoying problem solving. Important characteristics found for programming alone included cognitive and technical skills and being introverted. Important characteristics for programming in a group included interpersonal cooperation skills and personal maturity. Faculty and industry agreed on what constituted importance characteristics in both settings. Implications for programmer training and selection are discussed.

Keywords: Programmer success criteria, personality, critical thinking, student learning

1. PROGRAMMER SUCCESS CRITERIA

Research has identified characteristics that go beyond the general public's stereotypes of computer programmers. For example, Sitton and Chmelir (1984) conducted a study in which a group of computer programmers examined the Myers-Briggs Type Indicator (MBTI, see Table 1) in order to predict how other programmers would respond to it. The group predicted ESTJ (extroverted, sensing, thinking, and judging), whereas the actual responses of programmers turned out to be ENTP (extroverted, intuitive, thinking, and perceiving), with the majority being thinking perceiving types. However, Bush and Schkade (1985) and Lyons (1985) found the majority of programmers to be thinking judging types. Other research has examined predictors of computer proficiency. Evans and Simkin (1989) meta-analyzed research conducted between 1972 and 1987. Although they determined cognitive style to be significant in predicting a person's ability to master computer concepts, they also stated, "that the task of finding effective predictors of computer proficiency remains unfinished" (p. 1326). Research among IT & MIS professions has also recognized characteristics of computer programmers, as a result of their programming fundamentals.

Table 1
Myers-Briggs Type Indicator Terminology

Term	Meaning
E – Extroverted	Expressive, External
S – Sensing	Observant, Facts
T – Thinking	Tough-minded, Logic
J – Judging	Scheduling, Structured
I – Introverted	Reserved, Internal
N – Intuitive	Introspective, Ideas
F – Feeling	Friendly, Emotion
P – Perceiving	Probing, Flexible, Open

One individual difference variable that has received a good deal of attention is self-efficacy (i.e., the belief that one is capable of carrying out necessary actions to meet desired goals) and related concepts. For example, Hill, Smith, and Mann (1987) found that high computer self-efficacy beliefs were associated with increased likelihood of adopting and using advanced computer technology.

Other researchers (e.g., Martocchio and Webster, 1992; Webster and Martocchio, 1992) have shown that computer playfulness (i.e., the tendency to show spontaneous, inventive and imaginative interactions with computers) is associated with higher computer efficacy beliefs. More recently, Potosky (2002) found that high computer playfulness individuals who performed well during training showed the highest post-training self-efficacy scores (compared to those low in playfulness or those who did less well during training). Beyond computer playfulness, it is unclear what other individual characteristics might lead to greater computer self-efficacy or programmer success.

Much attention has been devoted to ensuring that information systems (IS) coursework is relevant to the needs of business (e.g., Byers and Van Over, 1996). However, researchers have given much less attention to how well the personal and social characteristics of programmers fit the needs of industry. The matching of individual characteristics with organizational demands has a long history in industrial and organizational psychology (Kristof, 1996; Schneider, 1987). Some IS researchers have indirectly addressed the person-organization fit question. For example, Trower, Willis, and Dorsett (1995/1996) found that students intending to major in IS believed that they would graduate with a marketable skill that allows a balance between their technical and business skills. Sivitanides, Cook, Martin, Chiodo, and Landram (1995) found that IS professionals rated verbal communication skills (e.g., being able to inform, persuade, and instruct; to listen carefully and to interpret feedback) as being very important to job success.

2. OVERVIEW OF THE RESEARCH

Our research is similar to past research (i.e., asking programmers about other programmers). However, instead of focusing on personality directly, this study examines programmers using perceived success criteria. We gathered information from two groups of experts: those responsible for the training of student computer programmers and those responsible for the product generated by computer programmers. Faculty members are experts who select programmers into their departments, improve students' technical skills, and ultimately train programmers for industry. Members of the computer industry critique a programmer's code, promote worthy individuals, and select competent members for programming teams. Clearly, it is important that faculty and industry members have similar views regarding the characteristics of good programmers. Consequently, our research attempts to identify the important characteristics that successful programmers show and whether faculty and industry members agree on those criteria for programming success.

In Study 1, we identified important programmer characteristics through expert interviews. In Study 2, faculty members and industry representatives rated those characteristics for importance in both individual and group settings. We expected that both faculty and industry members would agree on a core set of important programmer characteristics. In addition, we were interested in exploring potential areas of disagreement between faculty and industry as well as between individual and group settings.

3. STUDY 1

3.1 Participants

Ten faculty members (8 males, 2 females) from a large (20,000+) public university were interviewed to determine potential characteristics of successful programmers. Six participants were members of the computer science department and four were members of the computer information systems department.

3.2 Procedure

For each interview, two programming environments were described: programmers working on an independent project (alone) and programmers working within a group setting (group). Faculty members received the following instructions:

> Please describe, in your own words, what characteristics, skills, or attributes a successful computer programmer should possess, when working alone on an independent project (or, *when working as part of a group)*. Give no regard to prioritizing these attributes; simply discuss ideas as you think of them. Please elaborate as much as necessary to convey your ideas.

Each participant authorized a tape recorder to be used during his or her interview.

3.3 Results

Upon completion of all interviews, each distinct attribute relevant to programming success was coded from the tapes. We transcribed each characteristic onto an index card, with a total of 27 cards generated for the alone condition and 46 cards for the group condition. Some of these cards contained redundant attributes, though each was uniquely worded. For example, one card contained “doesn’t thrive on companionship” while a different card contained “doesn’t mind being alone.” These redundancies were included in order to maintain the broad range of expressions from the interview process. Three raters with appropriate professional qualifications independently sorted the cards into potential categories. Two raters were psychology faculty members and the third rater was a graduate student in psychology and computer science. Cards were retained if two out of the three raters grouped them in the same category. All other cards were eliminated. For example, consider the following hypothetical ranking for group #1, shown as [rater]{cards}: [A]{1, 4, 7}, [B]{1, 7}, [C]{2, 7}. This would result in cards 1 and 7 being placed within group #1, while cards 2 and 4 would be eliminated. The final categories were then analyzed and labeled.

In all, there were seven categories representing the alone condition (25 items) and seven categories for the group condition (24 items). Table 2 shows the categories (alone and group conditions) and the individual success attributes. Both conditions shared the categories of creativity, enjoying problems, and being conscientious. As the table indicates, technical and cognitive skills seemed to be very important for programming alone, whereas maturity and the ability to work with other people were important for programming in a group. All of the items from all of the categories were used for Study 2.

4. STUDY 2

4.1 Participants

Twenty university faculty members (15 males, 5 females) from computer science (16) and CIS (4) departments participated in the study. All of the instructors were actively teaching computer programming.

We contacted nearby departments by mail with a brief description of the study and an invitation to participate. None of the participants overlapped with those from Study 1. Faculty reported a mean number of years at their current position of 13.65 (S.D. = 10.01) and a mean of years writing code of 24.80 (S.D. = 11.00).

In addition, nineteen industry members (14 males, 5 females) working in programming fields participated in the study. Each industry member worked for a software company that was located using the local chamber of commerce directory. Industry respondents reported a mean number of years at their current position of 5.92 (S.D. = 4.63) and a mean of years writing code of 12.66 (S.D. = 10.86).

4.2 Procedure

Participants received the survey items from Study 1 in both group and alone conditions. Order of condition was randomized across participants (i.e., some received alone followed by group, others received group followed by alone). Instructions for the surveys were as follows:

> Please rate each item based on how important you feel the skill to be for a successful computer programmer working alone (with sole responsibility for a project) [or, *working as part of a group (such as a design team)*]. Consider a successful programmer to be someone who is favored by managers and who provides desired outcomes.

Participants rated each item on a 5-point scale (1 = *not at all important*, 5 = *extremely important*). We also invited respondents to add any important characteristics they thought were missing from the surveys.

4.3 Results

Because of the number of statistical tests conducted, we used an alpha level of .01 for all analyses. We conducted a series of one-sample *t* tests to determine whether or not faculty and industry members found the survey items to be important criteria for successful programming. A scale value of 4 ("very important") was used to determine the most important items per survey. A series of one-sample *t* tests (combining faculty and industry) indicated that ten of the alone items were rated significantly higher than 4 (see Table 3). These items came mainly from the cognitive skills and conscientious categories. Similar tests indicated that six of the group items were rated significantly higher than 4 (see Table 4). These items came mainly from the interpersonal cooperation category.

A series of independent sample *t* tests indicated that faculty members agreed with industry members regarding the characteristics of successful programming. Faculty responses did not differ significantly from industry responses for any items in either condition (alone or group). Similarly, male and female respondents agreed with each other regarding the success characteristics, differing on only one item from either condition (which would be expected by chance alone). Finally, very few participants added success characteristics to their surveys. Industry members included “pride,” “understand the user,” “attitude,” “willing to study at home,” “able to explain/justify,” “strong desire for learning,” “creative talent,” and “writing skills.” Faculty members added “modeling skills (spatial/math),” “calmness,” and “enjoys music.”

5. CONCLUSIONS

The purpose of our exploratory research was to identify success criteria for computer programmers working alone and in group settings. Despite potentially low statistical power due to the high number of *t*-tests conducted and the small sample sizes used, we established a baseline of common and unique characteristics for success in these settings. In addition, faculty and industry members agreed that the characteristics we identified were indeed important to programmer success. One strength of this research is that we used participants from both CIS and CS departments. Previous research of a similar nature tends to consider one or the other perspective rather

Table 2
Categories and Survey Items for Alone and Group Programming Conditions

Alone Condition	Group Condition
Cognitive Skills A mind of planning Follow standards (templates) Problem solving Attention to detail Breakdown of problem into pieces Analytical ability Information gathering, absorb relevant info Good interpretation Logical thinker	Interpersonal Cooperation Listens to others' ideas Helps others Interpersonal skills Open minded Open to criticism Capable of working with other people Communication Not always have to be in charge
Technical skills Loves to work with computers Technical skills Mathematics Breadth of computing	Maturity Secure Self-confidence Flexible personality Maturity
Introversion More introverted Doesn't mind being alone Self-disciplined to work alone	Enjoys problems Problem solving Enjoys puzzles Likes logic problems
Creativity Creativity Good reader Ingenuity	Creativity Creativity Ingenuity Imagination
Conscientiousness Persistence Determination	Analytical Breaks problem down into pieces Logical thinkers
Enjoys problems Enjoy puzzles Likes logic problems	Conscientiousness Determination Persistence
Planning Pace yourself Meticulous	Detail oriented Not mind detail work Meticulous

Table 3 - Descriptive Statistics for the Alone Condition Survey Items

Survey item	*M*	*SD*	*t*[b]
Problem solving	4.72	0.45	10.14*
Determination	4.58	0.68	5.39*
Persistence	4.53	0.64	5.19*
Analytical ability	4.50	0.64	4.94*
Attention to detail	4.47	0.55	5.42*
Logical thinker	4.40	0.59	4.28*
Breakdown of problem into pieces	4.38	0.70	3.37*
Technical skills	4.30	0.76	2.50*
Self-disciplined to work alone	4.30	0.97	1.96
A mind of planning[a]	4.28	0.69	2.57*
Information gathering	4.25	0.59	2.69*
Loves to work with computers	4.18	0.87	1.27
Likes logic problems[a]	4.08	0.87	0.55
Good interpretation[a]	4.05	0.69	0.47
Ingenuity[a]	3.95	0.76	-0.42
Creativity	3.93	0.86	-0.55
Meticulous[a]	3.90	0.75	-0.85
Breadth of computing	3.63	0.81	-2.94*
Good reader	3.53	0.93	-3.22*
Enjoys puzzles[a]	3.51	1.07	-2.84*
Follow standards (templates)	3.38	0.93	-4.27*
Pace yourself	3.35	0.98	-4.22*
Mathematics	3.25	0.93	-5.12*
Doesn't mind being alone	2.85	1.19	-6.12*
More introverted[a]	2.00	1.17	-10.68*

Note. N = 40, except where noted. [a] N = 39. [b] Test value of 4.00. * $p < .01$.

Table 4 - Descriptive Statistics for the Group Condition Survey Items

Survey item	*M*	*SD*	*t*[b]
Listens to others' ideas	4.60	0.63	6.00*
Capable of working with other people	4.58	0.50	7.26*
Communication	4.53	0.51	6.57*
Problem solving[a]	4.46	0.64	4.49*
Logical thinkers	4.40	0.63	4.00*
Open-minded	4.28	0.64	2.72*
Does not always have to be in charge	4.22	0.73	1.94
Breaks problems into pieces	4.22	0.77	1.85
Interpersonal skills	4.15	0.70	1.36
Open to criticism	4.05	0.81	0.39
Persistence	4.05	0.88	0.36
Determination	4.03	0.77	0.21
Like logic problems[a]	4.03	0.90	0.18
Helps others	4.00	0.75	0.00
Does not mind detail work	4.00	0.85	0.00
Flexible personality[a]	3.95	0.76	-0.42
Meticulous	3.93	0.76	-0.62
Self-confidence	3.90	0.74	-0.85
Maturity	3.83	0.87	-1.27
Creativity	3.80	0.76	-1.67
Ingenuity	3.80	0.72	-1.75
Secure[a]	3.79	1.00	-1.28
Imagination	3.50	0.72	-4.42*
Enjoy puzzles[a]	3.31	1.13	-3.84*

Note. *N* = 40, except where noted. [a] *N* = 39. [b] Test value of 4.00. * $p < .01$.

than both. In addition, whereas much research focuses on systems or software technical matters (Glass, Vessey, and Ramesh, 2002), our research focuses more on personality and behavioral factors that might relate to the successful building and implementing of systems. To what extent can these findings lead to improvements in training programs and selection methods within both academic and industrial domains?

There are similarities and differences among the success requirements for individual and group programming work. According to our respondents, a successful individual programmer has strong technical, cognitive and problem-solving skills, and is conscientious and creative. A successful team programmer is skilled at interpersonal interaction and cooperation, shows self-confidence and maturity, and is creative, analytical, and conscientious. To some, these attributes might suggest that programmers need to possess super-human skills in order to be successful. However, we prefer to take a broader view that these are characteristics that will increase the likelihood of programmer success rather than being absolutely necessary for success. Even if educators need to focus primarily on technical training, they might still want to highlight the importance of cognitive, problem-solving and other non-technical skills for their students. In her recent study, Potosky (2002) showed that performance during training interacted with a stable individual difference variable (computer playfulness) to predict post-training confidence and self-efficacy.

While classifying computer programming as either an individual or a group task was convenient for this research, industrial programming actually occurs under both conditions. Typical programmers will be required to write code as part of a project team while also working individually. Thus, it is likely that a programmer who possesses both sets of characteristics would be best equipped for work in industry. Unfortunately, being more of an introverted person might not be best for the interpersonal demands of group settings, and having strong interpersonal skills might not be very beneficial for individual programming tasks.

One potential direction for future research would be to consider the longer-term development and advancement of programmers within industry. For the rating task in Study 2, we did not specify whether the targets were entry- or higher-level programmers. On the one hand, faculty respondents may have thought more about entry-level skills, because that is what they work with more often. On the other hand, industry members may have had higher-level skills in mind when they did their ratings, if that is what they work with more often. Even if this was true, the similarity of our results across both groups suggests that the characteristics we identified apply to entry- and higher-level positions.

Of course, it is still possible that management-level positions require different characteristics for success than do entry-level jobs. In a survey of recent CIS graduates, Doke and Williams (1999) found that necessary knowledge and skills depended upon one's job classification. For example, for entry-level positions, technical programming skills were most important, whereas organizational knowledge and skills increased in importance for those in managerial positions. Interestingly, skills in systems development and interpersonal communication (e.g., speaking and writing skills) were important for all job levels. It would be useful to study the extent to which the success characteristics identified in our research apply to management level positions. A variety of psychological assessments are available to measure characteristics identified by our research. For example, the Wonderlic Personnel Test and Scholastic Level Exam can be used to measure reasoning and problem solving (Wonderlic, 1999), and the Teamwork-KSA Test can be used to determine skills such as collaborative problem solving, communication, and performance management (Stevens and Campion, 1994).

Another interesting research question is whether good programmers (or those possessing the characteristics of successful programmers) are likely to be attracted to programming careers and the extent to which they learn the important characteristics through their coursework and training experiences. In other words, are successful programmers "born" or "made?" Most likely, it is a combination of intrinsic interests, personal characteristics, and relevant experiences that lead to programmer success. VanLengen and Maddox (1990) found that instruction in computer programming did not improve critical thinking and problem-solving skills. If this is still true today, it suggests that some degree of self-selection among successful programmers may be occurring

It is encouraging that faculty and industry members agreed on the importance of the success characteristics we identified. It is not surprising that there was such agreement. After all, many of the faculty in our samples spent significant amounts of time in industry prior to becoming faculty. Whereas this agreement suggests that both groups are on the same page when it comes to training successful programmers, it does not indicate the degree to which these characteristics are cultivated or enhanced in school. Future research might examine the extent to which these characteristics can be affected through educational experiences or are addressed in existing curricula. Whereas it is unlikely that college courses can help students become more introverted, it seems reasonable to assume that cognitive, technical, and problem-solving skills, such as those identified in this study, can be taught. Similarly, teamwork experiences can provide students with opportunities to develop interpersonal cooperation skills, although there is no guarantee that such skills will be learned merely by participating in group projects. At the very least, instructors could provide these success criteria to their students so that they are aware of the demands, requirements, and expectations they may be facing when they graduate.

While this study explored the criteria of individuals, previous research has covered programmers within groups, such as expert programmers compared to novice programmers (Hoc, Green, Samurcay, and Gilmore, 1990) and programming team structures (Mantei, 1981). A fruitful direction for future research might be to compare expert and novice programmers to test the applicability of the current success criteria in advanced programming settings.

Focusing on the characteristics of successful programmers has the potential to improve the selection methods used to fill vacancies among project teams, alter the advising and selection criteria for academic programs, and change stereotypical views of the programming population. Our research offers initial information that might be used for these purposes.

6. ACKNOWLEDGEMENTS

We wish to thank Glenn Littlepage for his help with Study 1, and Charles Apigian, Jeff Clark, and the reviewers for their comments on an earlier version of this manuscript. Please direct correspondence to either author.

7. REFERENCES

Bush, Chandler M. and Lawrence L. Schkade (1985), "In search of the perfect programmer; which personality traits point people toward success in the dp industry?" Datamation, Vol. 31, No. 6, pp. 128-132.

Byers, C. Randall, and L. David Van Over (1996), "What to teach in an information systems curriculum: Depends on whom you ask!" Journal of Information Systems Education, Vol. 8, No. 2, pp. 40-45.

Doke, E. Reed and Susan R. Williams (1999), "Knowledge and skill requirements for information systems professionals: An exploratory study." Journal of Information Systems Education, Vol. 10, No. 1, pp. 10-18.

Evans, Gerald E. and Mark G. Simkin (1989), "What best predicts computer proficiency?" Communications of the ACM, Vol. 32, No. 11, pp. 1322-1327.

Glass, Robert L., Iris Vessey, and Venkataraman Ramesh (2002), "Research in software engineering: An analysis of the literature." Information and Software Technology, Vol. 44, No. 8, pp. 491-506.

Hill, Thomas, Nancy D. Smith, and Millard F. Mann (1987), "Role of efficacy expectations in predicting the decision to use advanced technologies: The case of computers." Journal of Applied Psychology, Vol. 72, No. 2, pp. 307-313.

Hoc, Jean-Michel, Thomas R. G. Green, Renan Samurcay, and David J. Gilmore (Eds.), (1990), Psychology of programming. San Diego: Academic Press.

Kristof, Amy L. (1996), "Person-organization fit: An integrative review of its conceptualizations, measurement, and implications." Personnel Psychology, Vol. 49, No. 1, pp. 1-49.

Lyons, Michael L. (1985), "The DP psyche; an international survey sheds new light on the personalities of data processors." Datamation, Vol. 31, No. 16, pp. 103-107.

Mantei, Marilyn (1981), "The effect of programming team structures on programming tasks." Communications of the ACM, Vol. 24, No. 3, pp. 106-113.

Martocchio, Joseph. J. and Jane Webster (1992), "Effects of feedback and cognitive playfulness on performance in microcomputer software training." Personnel Psychology, Vol. 45, No. 3, pp. 553-578.

Potosky, Denise (2002), "A field study of computer efficacy beliefs as an outcome of training: The role of computer playfulness, computer knowledge, and performance during training." Computers in Human Behavior, Vol. 18, No. 3, pp. 241-255.

Schneider, Benjamin (1987), "The people make the place." Personnel Psychology, Vol. 40, No. 3, pp. 437-453.

Sitton, Sarah and Gerard Chmelir (1984), "The intuitive computer programmer." Datamation, Vol. 30, No. 16, pp. 137-139.

Sivitanides, Marcos P., James R. Cook, Roy B. Martin, Beverly A. Chiodo, and Frank Landram (1995), "Verbal communication skills requirements for information systems professionals." Journal of Information Systems Education, Vol. 7, No. 1, pp. 38-43.

Stevens, Michael J. and Michael A. Campion (1994), "The knowledge, skill, and ability requirements for teamwork: Implications for human resource management." Journal of Management, Vol. 20, pp. 503-530.

Trower, Jonathan. K., G. W. K. Willis, and Dovalee Dorsett (1995/1996), "An evaluation of factors influencing intentions to major in information systems." Journal of Information Systems Education, Vol. 6, No. 4, pp. 206-215.

VanLengen, Craig A. and Cleborne D. Maddox (1990), "Does instruction in computer programming improve problem solving ability?" Journal of Information Systems Education, Vol. 2, No. 2, pp. 11-16.

Webster, Jane and Joseph J. Martocchio (1992), "Microcomputer playfulness: Development of a measure with workplace implications." MIS Quarterly, Vol. 16, No. 2, pp. 201-226.

Wonderlic (1999). Wonderlic Personnel Test & Scholastic Level Exam User's Manual. Libertyville, IL: Wonderlic, Inc

AUTHOR BIOGRAPHIES

Gregory D. Sterling is an Engineer with Jacobs Sverdrup, in Beavercreek, Ohio. His research interests include artificial intelligence, human computer interaction, and group dynamics.

Thomas M. Brinthaupt is Professor of Psychology at Middle Tennessee State University. His research areas include personality psychology, self and identity, and group processes.

Information Systems & Computing
Academic Professionals

STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the Journal of Information Systems Education have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

ISSN 1055-3096