PREDICTIVE VALIDATION OF A COMPUTER PROGRAMMER SELECTION TEST

THESIS

Presented to the Graduate Council of the
North Texas State University in Partial
Fulfillment of the Requirements

For the Degree of

MASTER OF SCIENCE

By

Sherman K. Duvall, B.A.

Denton, Texas

August, 1981

Duvall, Sherman K. Predictive Validation of a Computer Programmer Selection Test. Master of Science (Industrial Psychology), August 1981, 27 pp., 4 tables, references, 27 titles.

Subjects were 32 computer programmers employed in a large computerized tax-processing company in the Southwest. Ratings of each programmer's job performance by his/her immediate supervisor and scores on the Aptitude Test for Programmer Personnel (ATPP) were obtained.

Relationships between test scores and criteria were examined to identify significant ($p < .05$) correlations. Statistical treatment of data included zero-order Pearson product-moment correlation, multiple linear regression, and first-order semi-partial correlation analyses. Results indicated that the ATPP did not successfully predict ($p > .05$) the rated performance of the programmers.

TABLE OF CONTENTS

LIST OF TABLES

PREDICTIVE VALIDATION OF A COMPUTER PROGRAMMER SELECTION TEST

The ability to predict future job performance has been a primary concern of industrial organizational psychology. For years the premise has been held that the best predictor of future job performance is past job performance. However, past measures of job performance have not always been available, nor have previous measures often been relevant to future job performance, unless the jobs were highly similar. Therefore, the desire to find an alternate method for estimating future job performance has developed. Tests (i.e., paper-and-pencil and apparatus-type tests) have been implemented in all areas of academics and industry to assist in the systematic prediction of future success. With the increasing reliance on such tests, concern mounted as to how valid these tests were for the particular situations in which they were used. Thus, it has become apparent that a test should not be relied upon to estimate performance--present or future--unless it has been proven systematically to measure or predict what it purports to measure.

In an industrial setting, the necessity for valid predictors of performance is most important. Employers want to know whether an applicant will be successful in a particular position before they make a decision to hire. Quite a bit of research has been done in relation to selection methods in industry, but the study of specific predictors of performance in the area of computer programming has been relatively recent.

A wide variety of different tests have been utilized in screening applicants for computer programmer positions in business. Many of these tests were designed for purposes other than programmer selection. According to Palormo (1974), a major difficulty in this respect has been the widespread use of general aptitude tests "designed to yield measurements across the total population" (p. 1). A sizable portion of these types of tests have rarely been validated against performance criteria.

The United States Congress has been very succinct regarding its interest in validating tests such as those designed to predict job performance criteria. The 1964 Civil Rights Act, paragraph 703(h) of Title VII states:

> nor shall it be an unlawful employment practice for an employer to give and to act upon the results of any professionally developed ability test provided that such test, its administration, or action upon the results is not designed, intended or used to discriminate because of race, color, religion, sex, or national origin. (p. 23)

Title VII also states that tests used for employment purposes must have a "manifest relationship to the employment in question. . . . Professionally developed tests must be job-related" (p. 24). It is the employer's legal duty, therefore, to prove that any given requirement for employment is related to job performance. This qualifies the fact that tests, such as programmer selection tests, must be substantiated or validated as specific predictors of employment-related performance.

In a survey of programmer selection procedures, Watson (1961) sent survey questionnaires to 262 business firms in the United States and Canada. Of the 136 firms that replied, he found that 96 firms, or 71%, reported that they used psychological tests in programmer selection. A total of 29 different tests were used. Only two of these tests had been developed specifically for testing programmer aptitude. In a similar survey of 500 organizations, the System Development Corporation (SDC) conducted a National Survey of Digital Computing Personnel in 1962 (Perry, 1962). Of the 250 organizations that responded, 137 (57%) reported the use of tests in the selection of programmers and programmer trainees. These 137 organizations reported using more than 60 different tests in programmer selection. Perry found, however, that only three of these organizations had conducted validation studies, and that in each of these cases, a specially designed programmer aptitude test had been validated rather than an existing psychological or personnel test. As a result, only tests of a more specific nature have been shown to be effective, as well as legal, for the computer programmer population.

Aptitude testing has been an established, growing, scientific discipline within the broader field of applied psychology. Bower (1976) has made this comment on testing as a progressive science:

> The frontiers of knowledge advance; controversies rage and are settled; once-popular theories are disproved and discarded. The superstructure of the discipline is scientific experiment and analysis of empirical results. Its underpinning is mathematics, but not exclusively mathematical statistics. This is because the validation of tests--that is, the demonstration of their "job-relatedness," in the terminology of the *Griggs* and *Albermarle* cases--is often done by correlation analysis and similar statistical methods. As a result, like any scientific discipline, testing has a formidable jargon and a wealth of technicality among which the uninitiated must tread wearily. (p. 46)

Thus, when experimenters began developing a test for programmer selection, they had to consider the utilization of such scientific analysis.

In the early stages of development of tests designed especially to measure aptitude for programming, experimenters considered two things: (a) whether or not programmers actually differed significantly from nonprogrammers, and

(b) the extent and nature of such differences. Roemich (1963) used several aptitude tests to determine if computer programmers could be differentiated from nonprogrammers. He administered the Employee Aptitude Survey (EAS) and the Flanagan Aptitude Classification Tests (FACT) to 30 non-programmers, people with jobs other than programming, and 34 programmers. The performance of the two groups differed significantly on three of the EAS subtests--Numerical Ability (t = 3.40, p < .01), Symbolic Reasoning (t = 3.31, p < .01), and Numerical Reasoning (t = 3.25, p < .01); the two groups differed significantly on the Tables subtest of the FACT (t = 2.50, p < .01).

Perry and Cannon (1965, 1967, 1968) explored the area of programmer interests to see if they differed from the interests of people in other professions. They developed two programmer interest keys for the Strong Vocational Interest Blank--one for male programmers and one for female programmers. They found that, compared to men and women in other professions, computer programmers were less interested in people and more interested in mathematics and problem solving. They also found that scores on the programmer interest keys were positively related to satisfaction with the occupation. However, Perry (1968) observed that while dissatisfied programmers scored significantly lower on the keys than satisfied programmers, satisfaction was not related to salary progress in the field of programming.

In an attempt to identify the performance characteristics of successful programmers, Peres and Arnold (1963) asked 23 programmers and 6 programmer supervisors to describe, in a written essay, the behavior of the best programmer they had ever known. After a content analysis on the essays was performed, 140 statements describing programmer behavior were identified. These statements were sorted into 12 homogeneous groups and arranged in the form of a descriptive checklist. Each programmer and supervisor was given two programmer description checklists upon which he/she was asked to rate, on a 5-point scale, the best and the worst programmer he/she had ever known. By means of the Wherry-Winer method of factoring large numbers of items, the data obtained from the 58 checklists were subjected to a modified centroid factor analysis. From this analysis, Peres and Arnold found that six characteristics influenced programming performance. They were as follows.

1. Personal maturity and stability--The good programmer should have a stable and mature personality, as exhibited by the ability to work under pressure, to assume the responsibility for his own behavior, and to take suggestions and criticism in their proper perspective.
2. Cooperation in interpersonal relations--The successful programmer is a good listener.

He has the ability to work with, and to gain the cooperation of, other people.

3. Communication skills--The good programmer has the ability to speak and write clearly, and to understand and simplify technical terms.
4. Thoroughness and dependability--This factor characterizes the programmer who becomes thoroughly acquainted with the problem and attends to every detail before he/she begins to program; i.e., he/she has the ability to pay attention to relevant details while simultaneously maintaining a breadth of perspective regarding the problem as a whole.
5. Job interest and zeal--The successful programmer is one who is enthusiastic and innovative in his/her work. He or she has the ability to both accept and develop original and creative ways of programming.
6. Professional competence--The good programmer has the ability to analyze a problem from beginning to end, to reason and think logically, and to maintain technical proficiency. (pp. 88-89)

Rush (1961) similarly appraised the characteristics of a computer programmer. According to him:

> A successful programmer is quite intelligent with analytical, imaginative, and flexible thinking. He views each problem as a challenging exercise and attacks it with enthusiasm from several angles. Much of the successful programmer's time is spent defining the problem well so that he has all the details in mind when he begins to flow chart and code. Persistence is characteristic of the programmer in that he follows through until the problem is running efficiently. He gives attention to details but only to be sure that he has allowed for every contingency. (p. 41)

After a research review on the selection of computer programmers, McNamara and Hughes (1961) found that the ability to reason appears to be the most important single characteristic required of successful programmers. Other researchers have generally agreed that reasoning ability is an important factor relating to an individual's ability to program (Hollenbeck & McNamara, 1965; Palormo, 1974).

Two different types of tests which purport to measure reasoning have been designed specifically for testing programmer aptitude. One is an apparatus-type test and the other is a more traditional paper-and-pencil-type test.

Parenthetically, since apparatus-type tests are more difficult, time-consuming, and expensive to administer, they are seldom used in industry. Of the paper-and-pencil-type tests, the programmer aptitude tests have been exposed to more validation studies with programming performance than either the conventional psychological or the general personnel tests.

Several paper-and-pencil-type tests of programmer aptitude have been developed to predict programmer performance, the most popular of which is the Programmer Aptitude Test (PAT), constructed at IBM by McNamara and Hughes (1959). It has three parts: a number series, a figure analogy series, and an arithmetic reasoning series. When McNamara and Hughes (1961) studied the relationship between scores on the PAT and the technical job performance of 52 IBM 702 and 705 programmers, they found a correlation of .36 ($p < .05$) between the PAT and manager's ratings on a 5-point scale of technical performance.

Upshall and Riland (1961) discovered that supervisor's ratings of the job performance of 13 programmers at Eastman Kodak Company were significantly related ($p < .05$) to the PAT (Spearman rho of .61) and to the Brown-Carlsen Listening Comprehension Test (Spearman rho of .60). Rush (1961) found that the PAT scores of 161 programmers at the Standard Oil Company were significantly related to supervisor's ratings of learning ability ($r = .33$), of technical skills ($r = .28$), of imagination and ingenuity ($r = .37$), and of overall job performance (r = .31). Hence, it was apparent that programmer aptitude tests could be validated when the criteria used for validation procedures involved supervisory ratings.

Performance on a simulated work sample of programming has also been used as a criterion successfully predicted by the PAT (Howell et al., 1967). On two samples (N = 135; N = 118) of Civil Service employees at the United States Public Health Service, it was found that Part II of the PAT, the figure analogy series, and Part III, the arithmetic series, were the best predictors of a simulated work sample of programming out of four tests utilized. Howell et al. (1967) also found that in one of the samples (N = 118), the addition of the Numerical part of the *Federal Service Entrance Examination* (FSEE) to Part I and Part II of the PAT significantly improved prediction of the work sample performance (R = .71, N = 118; R = .60, N = 135).

The *Revised Programmer Aptitude Test* (RPAT) replaced the PAT in 1961, but the two forms are quite similar (r = .88). The RPAT was found to be significantly related (r = .44, p < .05) to supervisor's rankings of the overall job performance of 41 IBM 650 and 705 programmers (McNamara & Hughes, 1961). However, in 1964, two new but highly similar tests, the *Data Processing Aptitude Test* (DPAT) and the *Aptitude Test for Programmer Personnel* (ATPP), were developed by IBM to replace the PAT and RPAT (Hollenbeck & McNamara, 1965). The items contained in the DPAT and ATPP are very similar to those contained in the PAT and the RPAT; they are all paper-and-pencil tests of reasoning ability. However, the DPAT was designed especially for selection and placement within IBM, whereas the ATPP was designed primarily to aid in the selection problems of IBM's customers. The ATPP was made available for use by IBM customers and schools, but the DPAT was restricted to IBM internal use only. Published validation studies of these revised tests, particularly the ATPP, have been scant, but they have been both used extensively as if they had been thoroughly validated.

Although tests must be proven as predictive, problems related to validation studies have not been restricted to the tests. Jobs in the area of computer programming are technical and complex. Many ways have been tried by researchers to arrive at adequate job performance measures. Arvey and Hoyle (1974) worked on behaviorally based rating scales for systems analysts and programmers to determine the essential skills needed by personnel in entry-level data processing jobs. They found that task or job analysis was one way to obtain rater/ratee involvement which could possibly result in greater acceptance of the performance rating procedure. Several researchers have expressed the importance of rating employees on objective job-oriented traits (Buel, 1970; Heier, 1970; Miner, 1968). Sanders and Peay (1974) reported that "rating forms that use ambiguous trait names or descriptions or require ratings on traits which are not observable lead to unreliable results" (p. 33). They recommend that the traits to be evaluated "whether person-oriented, job-oriented, or both, should be determined by a thorough analysis of the jobs to be covered by the evaluation" (p. 35).

Another point worth mentioning was the disputed belief that validation studies actually had situational specificity. This idea had been based on the fact that there was considerable variability from study to study in raw validity coefficients even when jobs and tests appeared to be similar or essentially identical (Ghiselli, 1966). The explanation that developed for this variability was that the factor structure of job performance is different from job to job and that the human observer or job analyst is too poor an information receiver and processor to detect these subtle but important differences (Schmidt, Gast-Rosenberg, & Hunter, 1980).

The finding (Schmidt, Hunter, & Urry, 1976) that the typical validity study has only modest statistical power (perhaps in the neighborhood of .50) led Schmidt and Hunter (1977) to hypothesize that the observed evidence for situational specificity might be artifactual in nature. In developing this hypothesis, they postulated seven sources of artifactual between-study variance in observed validity coefficients: (a) sampling error (i.e., variance due to $\underline{N} < \infty$); (b) differences between studies in criterion reliability; (c) differences between studies in test reliability; (d) differences between studies in range restriction; (e) differences between studies in amount and kind of criterion contamination and deficiency (Brogdon & Taylor, 1950); (f) computational, typographical, and transcription errors (Wolins, 1962); and, (g) slight differences in factor structure between tests of a given type (e.g., arithmetic reasoning tests).

Using 14 distributions of validity coefficients from the published and unpublished literature for various tests in the occupations of clerical worker and first-line supervisor, Schmidt, Hunter, Pearlman, and Shane (1979) found that the first four artifactual variance sources noted above accounted for an average of 63% of the variance in validity coefficients, with a range from 43% to 87%. In Pearlman et al. (1980), these four artifacts accounted for an average of 75% of observed variance for 32 validity distributions based on job proficiency criterion measures and an average of 70% for 24 validity distributions based on measures of success in training. Thus, strong evidence has been developed that the observed variation in validities from study to study for similar test-job combinations has been artifactual in nature. These findings have cast considerable doubt on the situational specificity hypothesis (Schmidt et al., 1980).

However, in view of the limited available research on computer programmer selection tests, a study to identify and examine the relationship between certain predictors and computer programming performance criteria seemed appropriate. The purpose of this study was to investigate the relationships between the IBM Aptitude Test for Programmer Personnel (ATPP) and ratings of computer programmer job performance by supervisors. Specifically, it was proposed that the study examine the predictive validity of the ATPP. This was done by assessing the significance of the relationship between ATPP scores and supervisory ratings done at least 10 months after test administration. The programmers were hired between 1977 and 1979 at a large computerized tax-processing company.

Method

Subjects

Systems engineers (computer programmers) from a computerized tax-processing company in the Southwest were used as subjects in this study. The subjects (hereafter referred to as programmers) ranged in age from 21 to 38 years old. There were 12 male and 20 female programmers, which made a total sample size of 32. All programmers had a bachelor's degree with a small perdentage having some graduate education. The programmers were selected by personnel records available which contained aptitude test scores as well as performance evaluations.

Materials

The IBM Aptitude Test for Programmer Personnel (ATPP) was used as a predictor upon which this study was based. It consisted of three subtests, letter series, figure series, and arithmetical reasoning, and a total score.

The performance evaluations were based on supervisory ratings of five major criterion categories encompassing various technical as well as personal skills and abilities of programmers (see Appendix).

Procedures

Testing. Applicants for the position of systems engineer (computer programmer) were individually administered the ATPP. The test was administered by the employment administrator of the company in a small testing room. The period of testing ranged from May of 1977 to July of 1979. Each of the three subtests, letter series, figure series, and arithmetical reasoning, were timed (10 minutes, 15 minutes, and 30 minutes, respectively). There were four scores obtained for each applicant--three subtest scores and a total score. The subtest scores were based upon the total number of correct responses within each subtest. The total score represented the overall number of correct answers with 25% of the total number wrong subtracted. Any total scores with fractions of ½ or less were rounded down, whereas those with fractions greater than ½ were rounded up. The maximum score was 95; anyone with a score of 51 or above was considered for employment.

Performance evaluations. The performance evaluations were based upon ratings of programmers by their immediate supervisors. The criteria used in the ratings were produced from a job analysis conducted by the personnel manager. After familiarizing himself with the duties of a programmer at this particular company, he submitted a list of standards of performance to four supervisors. These standards of performance were categorized into five major groups: timeliness and accuracy; technical ability; self-improvement; initiative, self-reliance, and responsibility; and personal/professional skills. Each of these five categories was comprised of 5 to 8 standards for evaluating programming performance (see Appendix). Any of the performance standards that were not considered pertinent to successful programming by all four supervisors were eliminated. The programmer supervisors then agreed upon relative weights to assign each category according to how much each category should contribute to a composite performance evaluation (summary) rating.

Ratings on the performance standards within each of the five categories were obtained by assigning the following values: 1 = unacceptable; 2 = below average; 3 = average; 4 = above average; 5 = outstanding. After averaging the performance ratings for each category, a composite rating of the categories, utilizing the following weights, resulted: timeliness and accuracy (.30); technical ability (.20); self-improvement (.15); initiative, self-reliance, and responsibility (.15); and personal/professional skills (.20). The resulting rating was a single score having a value ranging from 1 to 5 (a continuum of unacceptable to outstanding, respectively). The performance evaluations included in this study were based upon supervisory ratings of the programmers' performance, according to the aforementioned standards, as of May 1, 1980, and May 1, 1981. The former (1980) rating was used for test validation; both ratings were used for criteria reliability estimation.

Correlations. Before the ATPP test scores were correlated with performance evaluation ratings, the subtest scores were transformed by means of subtracting 25% of the number of incorrect responses within each subtest. This was done to include the accuracy factor as well as the speed factor in the subtest scores (inasmuch as the subtests were timed). The three subtest scores were coded for computer analysis by variable names "IC," "IIC," and "IIIC," respectively. The total test scores were coded as "TOT." The five categories of performance criteria (ratings) were computer coded as "PE1," "PE2," "PE3," "PE4," and "PE5," respectively. The total or summary performance ratings were coded as "PET." Finally, tenure (months of employment) was coded as "TEN."

Having assigned variable names to tenure and the test (predictor) scores as well as the performance evaluation (criterion) ratings, an SPSS (Statistical Package for Social

Sciences) program was run on the data which: (a) performed Pearson correlations among all test predictor scores and performance ratings (including totals); (b) performed a multiple regression analysis of the three subtest scores with each performance evaluation rating (i.e., with PE1, PE2, PE3, PE4, PE5, and PET); (c) performed the aforementioned regression including tenure as a predictor; and, (d) performed a semipartial correlation analysis in which the effect of tenure on the summary performance ratings was partialled out. Intercorrelations were performed (Pearson _r_) among the performance evaluations as well to assess the degree to which they appeared to be independent measures.

To establish the reliability of the criterion measures, a test-retest correlational method was employed over an interval of 1 year. Reliability estimation was based upon a Pearson P-M correlation between the two groups of ratings performed by the same supervisors.

Results and Discussion

Means and standard deviations of all variables are computed for the 32 programmers. These data are presented in Table 1 (next page).

Table 1

Means and Standard Deviations of ATPP Scores, Performance Ratings, and Tenure

Test scores	Mean	SD
Letter series	29.45	6.15
Figure series	19.30	3.52
Arithmetical reasoning	17.73	4.43
Total	66.31	10.48

Performance ratings	Mean	SD
Timeliness and accuracy	1.23	.20
Technical ability	.78	.15
Self-improvement	.55	.10
Initiative	.65	.12
Personal/professional skills	.84	.15
Summary	4.05	.51

Tenure	Mean	SD
Months of employment	19.28	9.76

Note: *N* = 32

Pearson product-moment correlation coefficients between ATPP scores and performance ratings and between tenure and performance ratings are reported in Table 2. Inspection of Table 2 reveals that the four ATPP scores are not significantly correlated with the programmer performance ratings; however, tenure is significantly correlated with three of the six performance ratings (including the summary rating). Therefore, it appears from the data that the four ATPP scores do not significantly predict the rated performance of the programmers in any category. The correlations of tenure with technical ability, initiative, self-reliance, and responsibility, and the summary rating do attain statistical significance. This suggests that as the programmers gain experience on the job, they generally rate higher in technical ability and initiative, self-reliance, and responsibility, which contribute to a higher summary rating.

Table 2

Correlations of the ATPP and Tenure with Programmer Performance Ratings

	Performance ratings					
Test scores	1	2	3	4	5	Summary
Subtest 1	.15	.10	.10	.18	-.01	.15
Subtest 2	.09	-.22	-.06	-.26	-.14	-.14
Subtest 3	.21	.17	.27	.09	-.02	.21
Total	.22	.06	.15	.07	-.07	.13
Tenure	.14	.41**	.00	.51**	-.01	.29*

Note: N = 32.

*p = .05.
**p = .001.

In order to answer the question of whether an optimally weighted combination of the three ATPP subtest scores would significantly increase the validity of the test, a multiple regression analysis is performed. Summary statistics for this analysis are shown in Table 3. The F values obtained in this analysis are all low and nonsignificant ($p > .05$). Similarly, adding tenure to the regression analysis has a

Table 3

Results of the Multiple Regression Computed between Subtest Scores and Performance Ratings

	Performance ratings					
Subtest scores	1	2	3	4	5	Summary
R	.24	.39	.31	.46	.15	.36
F	.87	1.69	1.03	2.57	.36	1.39

Note: *N* = 32.

nonsignificant ($p > .05$) effect toward improving prediction of the criteria. However, when tenure is added to only the first subtest or the first two subtests, both resulting regression analyses are statistically significant ($p > .01$) toward predicting ratings of initiative, self-reliance, and responsibility. This, unfortunately, has no practical significance, since the test scores are used for selection purposes; tenure is not available as a predictor. Thus, according to these results, the optimal combination of the ATPP subtest scores do not significantly improve prediction of the criteria.

One final correlation is calculated between the ATPP total scores and the performance summary ratings. In this procedure, a semipartial correlation analysis is performed that correlates the ATPP total scores with that part of the performance summary ratings which did not correlate with tenure. The result is a correlation between TOT and PET of $r = .17$. This is an increase in the original zero-order correlation between TOT and PET (Table 2), but still nonsignificant (*p* > .05). This same condition occurs when IC and IIC are partialed from each other and respectively correlated with PET; both semipartials result in values greater than their zero-order counterparts. In this case it is due to the fact that IIC correlated negatively with PET, as can be seen in Table 2. The most reasonable conclusion extracted from these data is that the analyses contain several artifacts, (*N* = 32), restriction of range in the TOT scores and PE ratings, and possibly criterion contaminations.

A test-retest Pearson correlation is computed for the performance ratings to estimate their reliability. The results are shown in Table 4. These data indicate that only two of the individual performance ratings and the performance summary ratings are significantly reliable (*p* < .05). This

Table 4

Reliability of the Performance Ratings
from 1980 to 1981

Ratings	*r*
1	.27
2	.69**
3	.32*
4	.21
5	.26
Total	.64

Note: *N* = 31.

*p < .05.
***p* < .001.

shortcoming could be primarily artifactual in nature (small *N*, halo effect, lenience in ratings, scale attenuation, etc.).

Intercorrelations of the ATPP scores and the performance ratings result in some interesting findings. The letter series (IC) and the figure series (IIC) of the ATPP correlate significantly ($p < .001$) with each other in this study. Therefore, zero-order correlations of these with the criteria may be somewhat redundant. However, the use of only these measures in a multiple regression correlation successfully ($p < .05$) predicts PE4. As for criteria intercorrelations, PE1 correlates significantly with PE3 and PE4 ($p < .05$, $p < .01$, respectively); PE5 also correlates significantly with PE3 and PE4 ($p < .01$). Hence, there is a suggestion of halo effect in the ratings.

Strictly from the data of this study, the best criteria to use for correlational analyses with predictor measures are PE2, PET, or PE3, respectively. The most successful predictor-criterion relationship (taking predictor intercorrelations into account) is a linearly weighted combination of the first two subtests of the ATPP and PE4 (i.e., initiative, self-reliance, and responsibility). However, the reliability of PE4 makes this finding questionable. The best single predictor-criterion relationship is IIIC with PE3 (i.e., arithmetical reasoning with self-improvement; but this combination is statistically nonsignificant ($p > .05$). The ATPP lacks overall validity as a selection test in this study.

Summary and Conclusions

The purpose of the present study is to examine the validity of the ATPP in predicting the rated performance of systems engineers (programmers). A total of 32 programmers of both sexes employed at a computerized tax-processing company in the Southwest are used as subjects. The four ATPP scores serve as predictor measures. All subjects were administered the ATPP between May of 1977 and July of 1979, and subsequently hired within a few days from their respective test dates. The criteria of performance consists of five categorically weighted supervisory ratings and a summary rating. The ratings were performed on May 1, 1980, and May 1, 1981.

The method of study involves correlating the scores of each of the four ATPP measures with the six performance ratings using the Pearson product-moment correlation. In addition, a multiple regression analysis is performed to determine whether an optimally weighted combination of the ATPP subtests would significantly predict the performance ratings. Tenure is added to the regression analysis to ascertain the additional effect it would have as a predictor (concurrent). A semipartial correlation is also computed between the ATPP total scores and the performance summary ratings from which the effect of tenure has been removed Finally, a Pearson product-moment correlation is calculated between the 1980 and the 1981 performance ratings to estimate criteria reliability.

An analysis of the results obtained identifies no significant (p < .05) relationships between the ATPP scores and the performance ratings. Failure to obtain significant results should be interpreted while taking several factors into consideration: range restriction of the ATPP scores and performance ratings, sample size, criterion reliability, and other artifacts. The fact that the ATPP is no longer used in the employment department of the company in this study eliminates the possibility of any cross-validating procedures. The test has been replaced by another programmer aptitude test which is presently undergoing concurrent validation.

Although the primary objective of the present study is to investigate the predictive validity of the ATPP with performance ratings as criteria, there are some other interesting findings. Length of employment at this company tends to predict (concurrently) the programmer's performance ratings in technical ability, initiative, self-reliance and responsibility, and the summary. This is intuitively reasonable because such characteristics tend to improve with time. However, these findings have no practical significance for purposes of employment selection procedures.

The ATPP fails to render significant (p < .05) overall results as a selection device for programmers in a large computerized tax-processing company in the Southwest. This finding is based on correlations of the ATPP with supervisory ratings of programmer performance. Any interpretations of these analyses should consider the artifactual nature of the variance sources before drawing any conclusions.

Appendix

Performance Criteria

The candidate's performance will be determined by his/her success in:

1. Ability to complete projects and provide technical assistance timely and accurately (.30)

    - Able to prioritize and organize workload.
    - Notifies supervisor when deadlines are slipping.
    - Manages time wisely to work within external limitations.
    - Willing to work hours necessary to complete projects on time.
    - Controls and adapts to interruptions, changes, and disorders tactfully. Does not let others waste their time.
    - Responds to client and the company personnel's technical inquiries timely and accurately.

2. Technical ability (.20)

    - Understands tax law in order to analyze and understands accounting definitions and requirements of finished product.
    - Demonstrates grasp of PL1 and the company's system.
    - Pays attention to detail. Attempts to search out answers to program problems but knows when to ask questions.
    - Demonstrates ability to understand and grasp complex programming problems.

3. Self-improvement (.15)

    - Keeps informed of changes in area of expertise.
    - Expands knowledge of data processing.
    - Develops and presents to others new ideas and solutions.
    - Grasps new concepts, approaches, or systems.

4. Initiative, self-reliance, and responsibility (.15)

    - Takes action when appropriate on own initiative.
    - Asks for additional work when wlrkload is slack.
    - Knows when authority is being overstepped, obtains proper authorization.
    - Takes responsibility for projects from analysis, through testing to documentation.

5. Personal and professional skills (.20)

- Displays a positive attitude toward the company and takes pride in profession and work.
- Is courteous to clients and others contacted.
- Tactfully resolves conflict and/or problem situations with others.
- Cooperative and courteous.
- Considerate of other people's time.
- Resolves competing priorities and still maintains a good working relationship with those involved.
- Strives to improve from constructive criticism.
- Consistently adheres to reasonable hours of work as determined by manager.
- Observes approved policies and procedures.
- Attempts to resolve own personal conflicts with immediate supervisor. If still dissatisfied will discuss problems with manager.

References

Arvey, R. D., & Hoyle, J. C. A Guttman approach to the development of behaviorally based rating scales for systems analysts and programmer/analysts. Journal of Applied Psychology, 1974, 59, 61-68.

Bower, Catherine D., (Ed.). Test justification and Title VII. The Personnel Administrator, 1976, 21(1), 46-51.

Brogden, H. E., & Taylor, E. K. A theory and classification of criterion bias. Educational and Psychological Measurement, 1950, 10, 159-186.

Buel, W. Items, scales, and raters: Some suggestions and comments. Personnel Administration, 1970, 33(3), 26-30.

Ghiselli, E. E. The validity of occupational aptitude tests. New York: Wiley, 1966.

Heier, W. Implementing an appraisal by results program. Personnel, 1970, 47, 24-32.

Hollenbeck, G. P., & McNamara, W. J. CUCPAT and programming aptitude. Personnel Psychology, 1965, 18, 101-106.

Howell, M. A., Vincent, J. W., & Gay, R. A. Testing aptitude for computer programming. Psychological Reports, 1967, 20, 1251-1256.

International Business Machines. Manual for administrating and scoring the Aptitude Test for Programmer Personnel. White-Plains, New York: IBM Technical Publications Department, 1964.

McNamara, W. J., & Hughes, J. L. A review of research on the selection of computer programmers. Personnel Psychology, 1961, 14, 39-51.

Miner, J. Management by appraisal: A capsule review and current references. Business Horizons, 1968, 11, 83-94.

Nunnally, J. C. Psychometric theory (1st ed.). New York: McGraw-Hill, 1967.

Palormo, J. M. Computer Programmer Aptitude Battery. Chicago, Illinois: Science Research Associates, Inc., 1974.

Pearlman, K., Schmidt, F. L., & Hunter, J. E. Validity generalization results for tests used to predict job proficiency and training success in clerical occupations. Journal of Applied Psychology, 1980, 65, 373-406.

Perry, D. K. Vocational interests and success of computer programmers. Personnel Psychology, 1967, 50, 517-524.

Perry, D. K., & Cannon, W. M. Vocational interest of computer programmers. Journal of Applied Psychology, 1967, 51, 28-34.

Roemich, H. Testing programmer efficiency. Journal of Data Management, 1963, 1, 24-26.

Rush, J. M. A new look at programming aptitudes. Business Automation, 1970, 17, 36-45.

Sanders, M. S., & Peay, J. M. Employee performance evaluation and review: A summary of the literature. Naval Ammunition Depot Technical Report, 1974 (Aug.), RDTR No. 282.

Schmidt, F. L., Gast-Rosenberg, I., & Hunter, J. E. Validity generalization results for computer programmers. Journal of Applied Psychology, 1980, 65, 643-661.

Schmidt, F. L., & Hunter, J. E. Development of a general solution to the problem of validity generalization. Journal of Applied Psychology, 1977, 62, 529-540.

Schmidt, F. L., Hunter, J. E., Pearlman, K., & Shane, G. S. Further tests of the Schmidt-Hunter Bayesian validity generalization procedure. Personnel Psychology, 1979, 32, 257-281.

Schmidt, F. L., Hunter, J. E., & Urry, V. W. Statistical power in criterion-related validity studies. Journal of Applied Psychology, 1976, 61, 473-485.

U.S. 93d Congress, 2nd Session. Civil Rights Act of 1964, Title VII. Federal Register. Washington, D.C.: U.S. Government Printing Office, 1964.

Upshall, C. C., & Riland, L. H. An unpublished study at Eastman Kodak Company, 1958. Cited by W. J. McNamara and J. L. Hughes in A review of research on the selection of computer programmers. Personnel Psychology, 1961, 14, 39-51.

Watson, J. Programmer selection survey. Data Processing, 1961, 3, 9-13.

Wolins, L. Responsibility for raw data. American Psychologist, 1962, 17, 657-658.