AD 283 796

62-4-6

MEMORANDUM
RM-3245-PR
AUGUST 1962

# FACTORS IN SELECTING AND TRAINING PROGRAMMERS

Anders Sweetland

PREPARED FOR:
UNITED STATES AIR FORCE PROJECT RAND

The RAND Corporation
SANTA MONICA • CALIFORNIA

MEMORANDUM
RM-3245-PR
AUGUST 1962

# FACTORS IN SELECTING AND TRAINING PROGRAMMERS

Anders Sweetland

This research is sponsored by the United States Air Force under Project RAND — Contract No. AF 49(638)-700 — monitored by the Directorate of Development Planning, Deputy Chief of Staff, Research and Technology, Hq USAF. Views or conclusions contained in this Memorandum should not be interpreted as representing the official opinion or policy of the United States Air Force. Permission to quote from or reproduce portions of this Memorandum must be obtained from The RAND Corporation.

The RAND Corporation
1700 MAIN ST. • SANTA MONICA • CALIFORNIA

## PREFACE

A request from the Comptroller's Office, SBAMA, for assistance with the selection and training of programmers used in Electronic Data Processing work, provoked a reconsideration of a number of findings the author had uncovered in the programmer training program at the System Development Corporation. The findings should be of interest to those groups or organizations who employ programmers,* or are considering the development of an electronic computer installation.

Although the results presented in the Memorandum are preliminary, they do point both to the paucity of conclusive research on this important occupation and to the desirability of further studies aimed at the selection and training of programmers for all types of EDPE installations.

---

*It is necessary to understand that this study is concerned with programmers. Programmers are people who control the behavior of Electronic Data Processing Equipment (EDPE). Their activities are quite different from those people who control the behavior of Electric Accounting Machinery (EAM). (Also called Punched Card Accounting Machinery ... PCAM.)

## SUMMARY

This study attempts to determine some of the factors related to the selection and training of computer programmers.

Section I describes the evaluation of nine classes of programmer trainees according to their intelligence, motivation, and classroom performance. Supervisors' ratings were also obtained as a follow-up study. The findings show that both intelligence and motivation, particularly motivation, are good predictors of classroom performance. Intelligence is also a predictor of supervisors' ratings, but not as good a one as classroom showing. (The data did not permit testing the relationship between motivation and supervisors' ratings.)

The results described in Sec. I suggested that it would be profitable to explore non-cognitive measures. Section II deals with a study of one such measure: the vocational interest inventory. The investigation showed that programmers have interests that clearly distinguish them from the lay population. As a result, a scoring key for the Kuder Vocational Preference Record was developed. It is desirable to emphasize that this key should not be considered a final product, however, but rather as an illustration that such a key is feasible.

Section III discusses the potential fruitfulness of research in programmer characteristics, interests, and aptitudes; it suggests four areas for such research: (1) the organismic factors, with emphasis on characteristics other than intelligence, (2) programmer supervisors, (3) training, and (4) the working milieu.

## CONTENTS

## I. RELATIONSHIPS AMONG TRAINEES, INTELLIGENCE AND MOTIVATION

About 1954, the Numerical Analysis Department of the RAND Corporation felt the need for more formal selection procedures to meet their burgeoning demand for computer programmers. They approached the System Research Laboratory now the System Development Corporation (SDC) for assistance.

As a first step, members of the department underwent a battery of tests and were also ranked (subjectively) for programming ability. The tests were correlated with the rankings, and a multiple regression analysis made. (The multiple correlation coefficient was 0.59). The analysis yielded four tests that accounted for most of the explainable variance.

Thurstone Primary Mental Abilities (PMA) Test:

1. Verbal
2. Reasoning
3. Spatial

Thurstone Temperament Schedule:

4. E (emotional stability).

Both RAND and SDC have used this set to date, with minor variations. Both have dropped the E scale (which is awkward to explain and to administer). Both override the percentiles cut-offs if the applicant has compensating qualities (such as extensive background in mathematics, or programming experience). SDC requires that applicants have had at least one course in calculus.

Since the major components of most intelligence tests are measurements in the verbal-reasoning area, it is a reasonable conclusion that the two companies are limiting their hires to approximately the upper 2 to 5 per cent of the population in general intelligence (roughly, those with IQ's of 120 and up).

Although the use of the PMA is a little hard on the recruiters, the system works well: fewer than 5 per cent of terminations are due to lack of ability. As will be shown, however, intelligence is not the sole determinant of programming success.

Shortly after setting up the training program at SDC, we discovered that the PMA scores could not be used for counseling trainees: there was insufficient spread (as a result of the two 90th-percentile cut-offs used in hiring). It was impossible to determine the relative level of intelligence of the individual student. We desired a test that would:

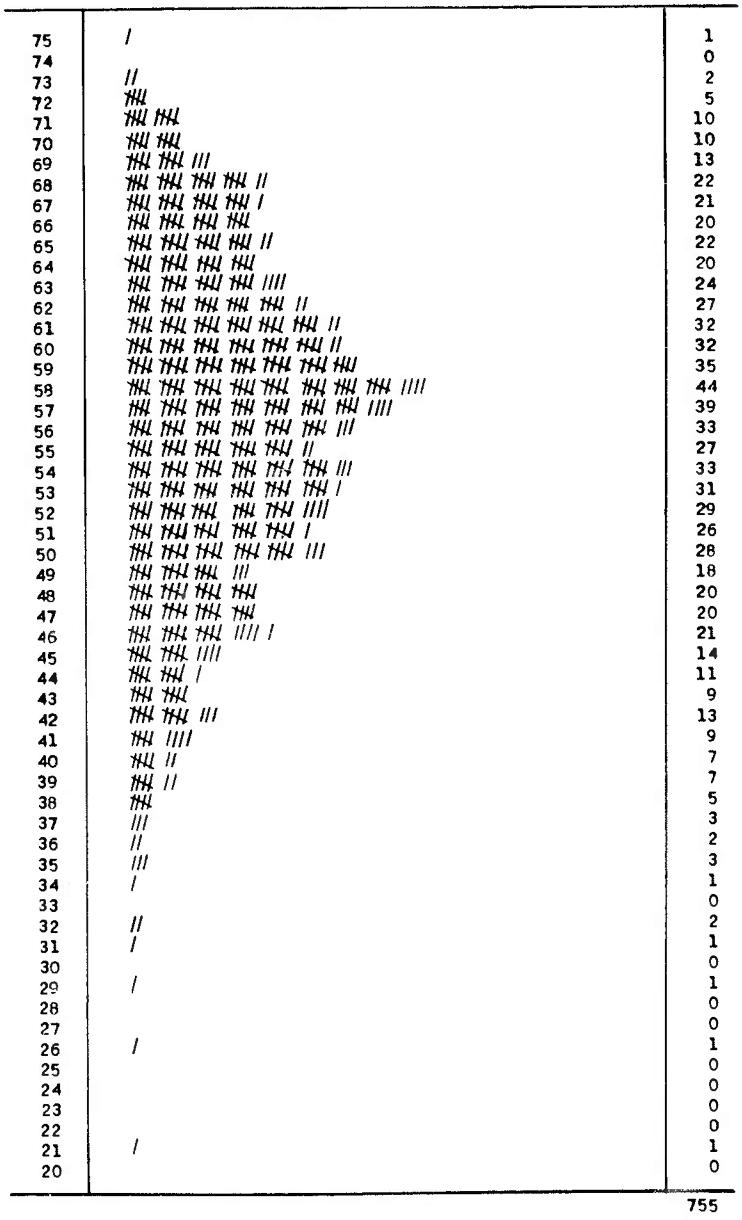
(1) Measure in the same dimensions as the PMA;

(2) Distinguish among trainees; and

(3) Be easy to administer and score.

The Otis-Higher Examination, Form D, met all three requirements. A factor analysis (coupled with other tests) showed that it measured the same dimensions. A reduction in testing time to 20 minutes yielded a spread from 25 to 75 in the raw scores. It could be administered to a group in less than 25 minutes, and scoring took less than a minute for each person tested (see Fig. 1). The measure of intelligence (unless stated otherwise) in the remainder of this study is the raw score obtained on the Otis Form D taken under a 20-minute time limit.

The programming classes were composed of approximately 20 students each. Each class ran eight hours a day and lasted eight weeks. As



M = 58.06

$\sigma$ = 8.57

N = 755

Fig. 1 -- Scattergram of Otis Scores (20-Minute Time Limit)

each unit of work was completed (lasting 3 to 5 days), the students were tested for knowledge in that unit. A constant program of item analysis was carried out. After the first four or five classes, the test items stabilized at the 50-per-cent level; i.e., each item was answered correctly by half the class. In the comparisons that follow, the measures of performance are based on the total scores for the eight-week training period. These will be either the raw scores, or the raw scores converted to rank-orders.

At the time, we were working on some characteristics of human motivation, and decided to include part of this work in the present study. Two estimates of motivation were used (with motivation defined here as a desire to learn programming):

(1) Estimates from instructors. Each class had two instructors. At the end of the course, each ranked the students independently. The lists were then compared and discrepancies resolved by discussion, which resulted in a third ranking. This third ranking was one of the measures used in the comparisons.

(2) Estimates by peers. Each student ranked the other members of the class (omitting himself). The rankings were first tested for stability, (using Kendall's Coefficient of Concordance*) and the composite rankings were rank-ordered. This composite ranking is the measure used in the comparisons. Because the peer estimates proved to be much more stable than the instructor estimates, we

---

*M. G. Kendall, Rank Correlation Methods, Charles Griffen & Co., Ltd., London, 1955.

Table 1

CORRELATION BETWEEN INTELLIGENCE AND CLASS PERFORMANCE
AND MOTIVATION AND CLASS PERFORMANCE

| | | Motivation | |
|---|---|---|---|
| Class | Intelligence: Otis & Grades | Inst & Grades | Peer & Grades |
| 3 | 0.31 | 0.34 | |
| 4 | 0.30 | 0.66 | |
| 5 | 0.59 | 0.82 | |
| 6 | 0.37 | 0.49 | 0.71 |
| 7 | 0.23 | 0.46 | 0.82 |
| 8 | 0.44 | | 0.74 |
| 9 | 0.30 | | 0.67 |
| 10 | 0.44 | | 0.57 |
| 11 | 0.60 | | 0.61 |
| Mean | 0.40 | 0.55 | 0.69 |

NOTE: Instructor and peer rankings were used as estimation of motivation. Kendall's Tau was used as the correlation measure. (Since the experiment began with Class 3, 1 and 2 are not included.) A Tau of 0.27 is significant at the 0.05 level for N = 20, the average class size.

finally used them exclusively.

The relationship among intelligence, motivation, and class grades is shown in Table 1.[*] In this comparison, the Otis scores and class grades were converted to rank orders. Kenall's Tau[**] was used as the statistical test. (To those more accustomed to the conventional Pearson product-moment, a Tau of 0.30 is about the same as a Pearson product-moment of 0.45).

---

[*] Since the time these data were collected, a number of studies relating intelligence and performance have been compiled. These are summarized in an excellent paper by Dallis K. Perry, Computer Programmer Selection Testing, Field Note 6371, System Development Corporation, Santa Monica, California, March, 1962.

[**] Kendall, *op. cit.*

The interesting part of these data is the ability of motivation scores to predict grades. In every instance, the estimations of motivation predict grades better than do the formal measures of intelligence. It thus appears that it would be profitable to investigate characteristics other than intelligence, a conclusion bolstered by the following observations.

At one time, as part of a salary review, each supervisor ranked his staff according to their "value to the company" and, this done, was asked to convert his ranking to a company-wide percentile ("Of all the programmers in the company, where would you place this person, percentage-wise?").

As a part of the quality improvement of the training program, we collected these supervisors' ratings. Admittedly, they are imprecise. They came from several points in time: sometimes shortly after the programmer was on the job, sometimes after he had been there several months. The supervisory experience was equally varied, nor was it possible to get 100-per-cent samples. Thus the supervisors' ratings lack the nicety of laboratory data, but at least they are real-world measures.

We matched the ratings with the Otis scores and training scores and correlated the samples (using Pearson product-moment). Since each sample (N = 50) is a mixture of several training classes, no direct comparison can be made with the previous data. See Table 2.

Two conclusions appear warranted:

(1) A student's showing during his training is important to an organization hiring programmers: it predicts his ratings by immediate

supervisors. This fact in turn implies that the training program per se is (or should be) a major concern of the organization.

(2) Programmer selection might be improved by investigating areas other than intelligence.

The Otis test does a reasonably good job of predicting classroom showing; and the classroom is a surprisingly good predictor of the supervisor's rating. The Otis test does not predict the supervisors' ratings very well, however.

Table 2

CORRELATIONS AMONG OTIS SCORES, TRAINING SCORES, AND SUPERVISORS' RATINGS

| | Otis-Grades | Grades-Ratings | Otis Ratings |
|---|---|---|---|
| | 0.46 | 0.66 | 0.26 |
| | 0.40 | 0.42 | 0.15 |
| | 0.25 | 0.48 | 0.01 |
| | 0.49 | 0.73 | 0.29 |
| | 0.47 | 0.32 | 0.27 |
| | 0.27 | 0.32 | -0.11 |
| | 0.40 | 0.54 | 0.29 |
| | 0.41 | 0.28 | 0.09 |
| Mean | 0.39 | 0.47 | 0.16 |

NOTE: A correlation of 0.27 is nificant at the 0.05 level for N = 50.

## II. AN INVESTIGATION OF A NON-COGNITIVE MEASURE

Section I suggested that exploration of the non-cognitive areas might improve the selection of potential programmers. During the interview, the recruiter faces one central question: whether to hire or not; to answer it, he needs objective measures that have predicted success in programming. Intuition does not help. For example, one would intuitively surmise that an applicant's mathematical background should foretell his success in programming. Experience proves otherwise. A sample of 40 cases was used as a test, in which semester hours of mathematics (which ranged from 6 to 40) were correlated with class grades. The correlation was -0.17 (not significant).

Vocational background was equally unfruitful. So was the amount of college that applicants had had: some of the poorest students were Ph.D's, but so were some of the best. As in most professions, there were few outstanding females, but females (again as in most professions) were similarly rare at the low end of the continuum. Other than the intelligence test, our fictive recruiter has little to help him at present.

These negative considerations lead to the following rationale. Most vocational guidance programs encompass three areas of measurement, two of which have been considered: (1) ability, (2) achievement, (3) interest. The intelligence test measures ability and does predict success in programming. The simple and obvious achievement measures (amount of college, amount of mathematics, etc.) yielded little.* Finally, the area of interest-measurement has not been explored at all. It is an appealing one for two reasons:

(1) A vocational interest score is believed to be a low-key measure of motivation; and

(2) The data of Sec. I indicate that motivation plays a major role in success in programming.

The following is a description of an exploratory study in vocational interest testing. The Kuder occupational preference record was given to 100 programmer trainees. Since trainees are not professional programmers, we had misgivings about using these data to establish a preference profile; however, it is comparatively easy to get data from trainees and very difficult to get data from professionals. Consequently, our first question was: Will trainees reasonably resemble professionals in their responses to the Kuder?

The hundred trainees were split randomly into two groups. A Kuder key was established by standard procedures. (Essentially the procedure was to ask whether the proportion of the experimental group who answered an item in a specific manner differed significantly from the normative group proportion. If so, the item was incorporated in the scale.) In this study, an item had to distinguish (from the normative group) at the 5-per-cent level in both experimental groups to be included.

This procedure established a scoring key for trainees. The next question was: Do professional programmers answer the Kuder in

---

*There is one unexplored area, however: success in medical school has been predicted by combining college grades in a specific set of related subjects. It may be that combinations of grades would have similar predictive value for programming success.

the same way trainees do? The answer sheets of the 100 students and of 30 RAND programmers were scored with the key constructed from the trainees' answers. In addition, 100 answer sheets from the general population were obtained and were scored with the same key after being split into two random groups. The results of this manipulation are shown in Figure 2.

The reader can see from Fig. 2 that trainees and professionals answer the Kuder in essentially the same manner, and that their interests (as measured by the Kuder) are quite different from those of the general population.

Following this happy discovery, we made several trial-and-error attempts to find a scoring key that would yield the least overlap among the three samples of programmers and the two samples from the general population. The resulting key was obtained by isolating those items that distinguished programmers from the lay population in all three of the experimental groups (i.e., to be included, an item had to distinguish in each of the three separate samples). The answer sheets were rescored with this key. It is obvious that programmers have interests different from those of the lay population. (See Fig. 3 for results.)

Two precautions are in order: first, although the area of vocational preference appears to be promising, there has been no validation for this key; and although programmers can be distinguished from the general population in terms of their interest, it remains to be demonstrated whether these interests are related to success in programming. (For example, if we partial out intelligence,

| | 1st 50<br>SDC | 2d 50<br>SDC | RAND | 1st 50<br>General Population | 2d 50 |
|---|---|---|---|---|---|
| 52 | | // | | | |
| 51 | // | | | | |
| 50 | | | / | | |
| 49 | // | / | | | |
| 48 | / | / | | | |
| 47 | // | / | | | |
| 46 | / | //// | | | |
| 45 | / | / | / | | |
| 44 | /// | //// | / | | |
| 43 | /// | /// | / | | |
| 42 | 𝍸 / | / | | | |
| 41 | //// | 𝍸 | / | | |
| 40 | 𝍸 | 𝍸 / | // | | |
| 39 | //// | 𝍸 | // | | |
| 38 | / | /// | //// | | / |
| 37 | / | /// | 𝍸 // | | / |
| 36 | 𝍸 | // | | // | / |
| 35 | /// | | /// | / | 𝍸 |
| 34 | / | // | 𝍸 | | /// |
| 33 | // | //// | | / | / |
| 32 | | / | / | /// | / |
| 31 | / | | / | // | //// |
| 30 | / | | | // | / |
| 29 | | / | | | //// |
| 28 | | | | / | //// |
| 27 | | | | // | 𝍸 / |
| 26 | / | | | // | /// |
| 25 | | | | //// | 𝍸 |
| 24 | | | | //// | //// |
| 23 | | | | 𝍸 /// | /// |
| 22 | | | | / | /// |
| 21 | | | | //// | |
| 20 | | | | / | |
| 19 | | | | //// | |
| 18 | | | | //// | |
| 17 | | | | // | |
| 16 | | | | | |
| 15 | | | | // | |
| M = | 40.26 | 40.24 | 37.66 | 24.20 | 28.64 |
| $\sigma$ = | 5.06 | 5.01 | 3.96 | 5.41 | 4.45 |
| N = | 50 | 50 | 30 | 50 | 50 |

Fig. 2 — Kuder Scores, First Trial: Scattergrams of Three Programmer and Two General Population Samples

| | 100 SDC + 30 RAND | 100 Norm |
|---|---|---|
| 50 | | |
| 49 | | |
| 48 | | |
| 47 | | |
| 46 | | |
| 45 | // | |
| 44 | / | |
| 43 | | |
| 42 | 卌 | |
| 41 | //// | |
| 40 | /// | |
| 39 | 卌 卌 /// | |
| 38 | 卌 卌 // | |
| 37 | 卌 卌 卌 /// | |
| 36 | 卌 卌 / | |
| 35 | 卌 卌 /// | / |
| 34 | 卌 //// | // |
| 33 | 卌 卌 卌 | |
| 32 | 卌 卌 | 卌 |
| 31 | //// | //// |
| 30 | // | 卌 卌 / |
| 29 | //// | 卌 / |
| 28 | /// | 卌 / |
| 27 | | //// |
| 26 | / | 卌 卌 |
| 25 | | 卌 // |
| 24 | | 卌 // |
| 23 | | 卌 卌 |
| 22 | | 卌 |
| 21 | | 卌 //// |
| 20 | | 卌 / |
| 19 | | / |
| 18 | | // |
| 17 | | /// |
| 16 | | / |
| 15 | | |
| | M = 35.684<br>σ = 3.646 | M = 25.410<br>σ = 4.338 |

Fig. 3 — Final Kuder Key Scoring, Comparing 130 Programmers with 100 General Population

will the interest scores be related to success in training, success on the job, or continuation in the profession of programming?)

Until the key has been validated, it is desirable to follow the more conservative approach used in the VA program (Public Law #16 and Public Law #346) following World War II. In the VA program, unless a person showed a career interest above the 75th percentile of the general population, it was deemed that his interests were not crystallized enough to warrant his planning for that career and he was encouraged to explore further. (We may note in passing that a 75-per-cent cut-off of the general population would have cut off the lower 7 per cent of the programmers used in this study.)

Second, unlike intelligence (which is fixed at conception), interests are malleable. It is generally true that a person's vocational interests tend to stabilize when he is about twenty-five, but this is largely a cultural happenstance. A person can become interested in a new profession at the age of 50. This is another way of saying that vocational guidance, based on interest testing, should operate only in the crudest of dimensions: a man with an interest in science and mathematics and none in social services should presumably be steered toward the engineering and related professions, where he is more likely to find a stimulating occupation, and away from the social-service professions. Because interests are learned, however, it would be incorrect to assume that such a man cannot become interested in social services.

To summarize this Section: Programmers appear to have a set of homogeneous interests that distinguish them from the general population, and the selection of programmers might be improved by measuring these interests. Interest scores should be used conservatively, however, until interest is shown to be a valid predictor of programming success. If interest measures are to be used, it is suggested that the upper 25 per cent of the general population be used as a recruiting cut-off point.

## III. SUGGESTIONS FOR FURTHER RESEARCH

The explosive growth of computer technology in the past ten years indicates that the profession of programming warrants considerable exploration. Computer installations are expensive, and what we get out of them is highly related to the talents of the programming staff. It is worthwhile to begin exploring the dimensions of programming talent.

The programming activities of most large computer installations fall into two broad categories:

(1) Service functions. These are the bread-and-butter jobs: payroll, billing, insurance premium computations. Essentially, external needs define the programmer's actions. The jobs generally make considerable demands on his ingenuity, but rarely necessitate inspired, creative imagination.

(2) Research in programming. In contrast to the bread-and-butter tasks, basic research is another world. In it, people try to get the computer to do things Turing never had in mind, such as translations of language, learning, concept formation, problem-solving (in the psychological sense), and abstract writing. Here imagination is a *sine qua non*.

Intuitively, we would surmise that these two types of activities demand two different types of programmers, although we do not know what their distinguishing qualities may be. (We may be wrong, however: perhaps the same programmer can double in both types of activities.) In one instance, the program is a means to an end (as statistics are used in psychological experiments); in the other the program is an end in itself (e.g., the development of linear programming). Research on this question is advisable.

A second area for exploration is determination of the qualities of good programmer supervisors. From a company's point of view, good supervision is critical; but how does the recruiter respond to the request, "Of the ten programmer trainees you recruit, include two who are potential supervisors?" More basically, what qualities (if any) distinguish good programmers from good supervisors?

A third area for research is training. One question is how old (or young) a person should be when he begins to learn programming. Some of the work on concept formation suggested that the ability to form concepts is independent of chronological age but directly related to mental age. The same may be true of programming. Extrapolating the Otis results suggests that the brighter (upper 5 per cent) juniors and seniors in high school could be trained in programming and benefit by the process, but that it would be an unusual high school freshman who could do as well. Psychologists recognize this as a variation of the maturation-versus-learning problem.

Finally, there is the comparatively unexplored area of the working milieu. If it is true, as our data imply, that the programmer is a different sub-species, perhaps he needs a special kind of environment in order to be productive. The work of Pelz* on the productive

---

*D. Pelz, "Some Social Factors Related to Performance in a Research Organization", Adm. Sci. Quart., Vol. 1, No. 3, December, 1956, pp. 310-325.

milieu suggests that the scientifically productive working environment differs dramatically from what one might guess (among other things, the environment the "organizational man" finds comfortable is the least productive of ideas). But Pelz's subjects (scientists) may be different from programmers. Again, further research is advisable.

## IV. CONCLUSIONS

A number of exploratory studies were made to determine what factors might be related to success as a programmer. The findings were:

(1) Both intelligence and motivation -- more notably, motivation -- are closely related to classroom performance in programmer training.

(2) Both intelligence and classroom performance are related to on-the-job ratings by supervisors. The stronger relationship of the two is that between classroom performance and supervisor ratings.

(3) Programmers have interests that clearly distinguish them from the lay population. A tentative Kuder vocational preference scale has been developed.

Research in programming promises to be rewarding. The study suggests four areas: (a) predictive characteristics of programmers (particularly non-cognitive measures), (b) supervisory qualities, (c) the working environment, and (d) training.