SCIENTIA
WNNR·CSIR

CONTRACT REPORT

C/PERS 168

COMPUTER PROGRAMMERS - A PRELIMINARY CRITERION AND TEST VALIDATION STUDY

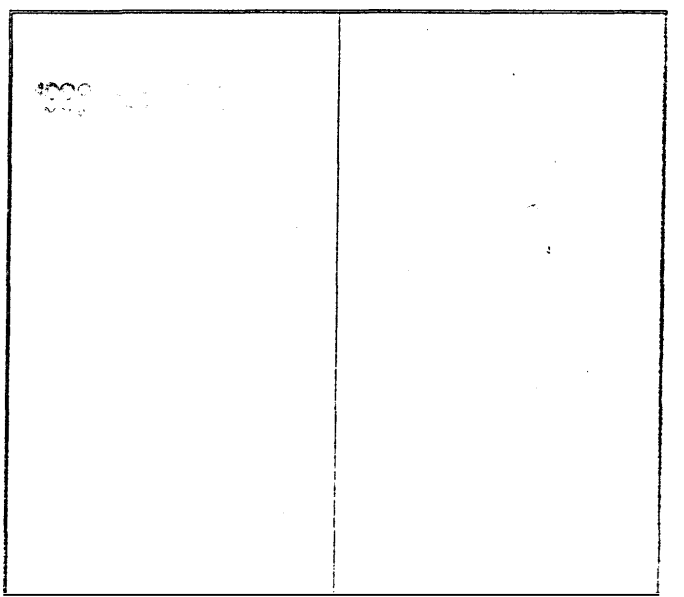
submitted to

THE COMPUTER SOCIETY OF SOUTH AFRICA

NATIONAL INSTITUTE FOR PERSONNEL RESEARCH
COUNCIL FOR SCIENTIFIC AND INDUSTRIAL RESEARCH

CSIR Contract Report, No. C/PERS 168 UDC 681.3.00-05:[658.311.5:001.891
Johannesburg, South Africa June 1969

COMPUTER PROGRAMMERS - A PRELIMINARY CRITERION AND TEST VALIDATION STUDY



NATIONAL INSTITUTE FOR PERSONNEL RESEARCH
COUNCIL FOR SCIENTIFIC AND INDUSTRIAL RESEARCH

ACKNOWLEDGEMENTS

This project was directed by Mr. D.J.M. Vorster.

The criterion schedule was developed by D.W. Steyn. He was assisted by E. Maughan-Brown and M. van der Merwe. R.S. Hall acted as advisor and co-ordinator.

The test battery was suggested by J.M. Schepers.

The analysis of data was undertaken by R.S. Hall.

The report was written by D.W. Steyn. R.F. Skawran, R.S. Hall and J.M. Schepers acted as advisors on the preparation of the report.

We wish to express our appreciation of the co-operation extended by the organization which participated in this study.

SYNOPSIS

A project was initiated in April 1968 by the N.I.P.R. in conjunction with the Computer Society of South Africa with a view to improving the selection of personnel in the data processing field.

The first phase of the project consisted of an extensive review of literature on selection and criterion problems as well as comprehensive job descriptions of operator, programmer and system analyst jobs. Findings of this investigation were reported by Van der Merwe (20).

The second phase was concerned with the development of a criterion and the validation of a number of tests. This part of the study was limited to programmers.

A criterion schedule consisting of 40 task statements was developed. The criterion conformed to metric requirments regarding reliability and validity. The statistical analysis was based on a sample of 60 experienced programmers in a large industrial organization.

A preliminary validation study was carried out on a subsample of 49 programmers. Promising results were found.

Unresolved aspects to be dealt with in the following stages are discussed.

CONTENTS

COMPUTER PROGRAMMERS - A PRELIMINARY CRITERION AND TEST VALIDATION STUDY

1. PURPOSE OF THE INVESTIGATION.

The purpose of the investigation was to develop a criterion of on-the-job proficiency for computer programmers working in commercial environments in South Africa and tentatively to determine the potential usefulness of a number of tests for the selection of programmers.

2. SCOPE OF THE INVESTIGATION.

A survey of literature revealed that one of the basic deficiencies of previous validation studies in this field had been inadequate criterion measures. The reliabilities of criteria were nowhere reported and the impression was gained that some basic considerations had been either ignored or overlooked. It was felt that the need existed for criterion measures which conformed to psychometric requirements without compromising practicability, relevance of behaviour or comprehensiveness.

Available instruments could not be adapted to suit this need and necessitated the development of a criterion schedule which measured a wide variety of on-the-job performances.

Criterion development and the subsequent test battery were extensively based on the job descriptions and recommendations of a previous study by Van der Merwe (20). As a supplementary procedure, a number of computer experts were interviewed to ensure the completeness and relevance of the criterion. A comprehensive survey of literature was also undertaken to ensure as broad and as eclectic an approach as possible.

The final version of the criterion schedule was applied to a sample of 60 programmers in one specific organization to minimise extraneous sources of variance, bearing in mind the large inter-organizational differences described by Van der Merwe (20). These programmers had all been recruited, selected and trained by the particular

organization and had similar duties and were subject to the same system of supervisory control.

A tentative test battery of approximately four hours' duration was administered to a sub-sample of 49 of these programmers, the remainder either having terminated their services or otherwise being unavailable for testing.

The limited time available narrowed the scope of testing to such an extent that not all hypotheses could be subjected to empirical confirmation. The tests used were an Intelligence Test, a Temperament Questionnaire, a test for problem-solving ability called the Concept Attainment Test and a test of the ability to integrate information. Certain domains such as career interest and inductive reasoning as well as certain clerical abilities could not be sampled, because of the limited time available.

3. CRITERION DEVELOPMENT.

In criterion development two types of criteria present themselves. The assessment of performance in a more or less formal training situation is known as a training criterion and usually consists of indices of proficiency such as marks attained in an examination or assessments by a teacher or instructor.

On-the-job criteria are concerned with the performance of an individual in a work situation and may consist of indices of proficiency in a real or a simulated task.

3.1 Training Criteria

A training criterion is essentially a short term criterion i.e. good training results are in many cases not predictive of future on-the-job performance. Studies by Severin (14), Taylor and Nevis (18) and Brown and Ghiselli (4) have demonstrated that in general, low correlations exist between training grades and subsequent on-the-job performance. Although these studies were not specific to programming, it is reasonable to assume that the same phenomenon would occur in programming - the reasons probably being the more complex demands of the

working situation and the fact that training is sometimes job-irrelevant. Stalnaker (16) had already pointed in this direction when he observed that methods of selection for training of computer programmers alone appeared to be doubtful; results had shown that border-line students quite frequently developed highly sophisticated programming skills once exposed to the actual working situation.

A training criterion is not without merit. Psychologists have been fairly successful in predicting training grades and where training is costly, short-term prediction becomes an economic proposition.

It was, however, felt that a training criterion could not be accommodated within the framework of this project. With the exception of formalized training at a few universities and technical colleges, training is still largely informal and undertaken by individual firms. This lack of uniformity precludes the formulation of comparable, meaningful criterion measures.

3.2 On-the-job Criteria.

A number of approaches may be pursued in the development of on-the-job criteria. Among these are merit ratings, objective recordings, simulated tasks and achievement tests. A survey of literature suggests that although these measures have been extensively used, little attention was paid to a number of relevant prerequisites for an acceptable criterion.

3.2.1 Prerequisites for an acceptable Criterion.

The most desirable prerequisites for an acceptable criterion can be summarized as follows:

(a) Practicability

One of the important considerations in criterion development is the recognition by the psychologist of the practical limitations imposed on him by the realities of business and organizational life.

The process of employee evaluation must be an economic and practicable proposition in terms of time and expenditure involved. Management or rater resistance is an ever-present stumbling block in the assessment of individual performance. The crux of the problem is the development of criterion measures which are simple to apply and practicable, yet scientifically rigorous.

(b) Relevance

The relevance of performance is another factor of importance. Each aspect of the criterion must be unambiguously related to usefulness and should reflect the utility of the individual as a working unit. The inclusion of criterion-irrelevant aspects of behaviour or performance only confuses the issue, introduces unnecessary bias and fails to reflect the true relationship between performance and utility.

(c) Comprehensiveness

A good criterion should sample relevant performance as comprehensively as possible. The omission of criterion-relevant aspects leads to a distorted or one-sided picture of the individual being assessed and also leads to the suppression of the relationship between predictive tests and on-the-job performance (10).

(d) Reliability

One of the most important characteristics of a sound measuring instrument is its ability to rank people in a consistent way in terms of the attribute being measured. A reliable instrument is consistent over time and displays little fluctuation when re-applied. Another characteristic of a reliable measuring instrument

is that different raters agree in their assessments of the same individuals.

The reliability of a rating criterion is a function of the following factors (10):

(i) The competency of the raters;

(ii) the simplicity of the behaviour being assessed;

(iii) the degree to which the behaviour is overt;

(iv) the opportunity to observe;

(v) the degree to which the rating task is defined.

To the above, the following may be added:

(vi) Criterion assessments should be based on many observations;

(vii) the unit of time should be adequate for a sufficient sampling of behaviour (13), i.e. allow for consistent measurement.

3.2.2 Different on-the-job Criteria.

In considering the development of on-the-job criterion measures, a number of alternative approaches present themselves:

(a) Situational Tasks

The possibility of developing a situational task consisting of a standardized programming task which could be scored by outside experts was investigated. It soon became apparent that this approach was not feasible. An artificial task samples only a limited domain of performance, due to the practical limitations of time. Furthermore, the costs in terms of time spent on the test and its subsequent scoring would have been

so prohibitive as to render it impracticable.

(b) Achievement Tests

An achievement test as a criterion was developed by Berger and Wilson (2), (3). Called the Basic Programming Knowledge Test it measures knowledge of six distinct programming areas viz. logic estimation and analysis, flow diagramming, programming constraints, coding operations, programme testing and checking, and documentation. Knowledge of the basic principles, techniques and applications of programming is tested irrespective of the computer or coding language used.

This criterion does not stand very close scrutiny. Van der Merwe (20) has pointed out that the basic assumption that knowledge equates with proficiency is open to criticism. It is also important to note that knowledge is frequently a function of training so that a considerable proportion of the variance of the criterion could be generated by different training methods.

(c) Trait Ratings

One of the more commonly used systems of personnel evaluation is the so-called trait rating system where, typically, such psychological traits as dependability, initiative and co-operation are assessed on either a five-point or a nine-point scale. Evaluation is also frequently done on such factors as quality of work, volume of work and job knowledge.

These trait ratings may serve a valuable purpose in many instances; they may serve as a basic document for personnel development, personnel counselling or as a basis for promotion

or other administrative action.

In general, these trait ratings have, however, dubious value as measures of on-the-job proficiency. Poor ratings may be a function of poor communication between rater and ratee (15) and the system may be further complicated by rater deficiencies such as inadequate training, stereotype formation and physical propinquity of rater and ratee. The system is furthermore particularly susceptible to the halo error; the rater tends to carry a certain frame of reference across traits.

In some firms care is exercised to minimize the above objections. Raters are trained and familiarized with the system and a trained psychologist attends evaluations where supervisors have to substantiate ratings with critical incidents.

The main objection against trait assessments is that the emphasis is placed on the individual's character and personality, and not on his work.

(d) Objective Recordings of Performance

In some instances programmer performance is carefully tabulated in terms of the total number of test runs needed to get a program working properly. Further objective measures may be the recording of the time needed to complete a program.

Where programmers are assigned specific individual tasks, such objective indices may serve a valuable purpose. There are, however, limits to the usefulness of these indices for test validation. No two programs are identical in terms of complexity, skill demanded and operating

requirements so that no common yard-stick is available. It is further difficult to specify the period of time over which this information should be gathered to ensure consistent results. In general, several studies have shown that the relationship between these objective indices and supervisory assessments tends to be low (19). It can thus be argued that objective indices do not meet the requirement of comprehensiveness. A further objection is, that for the purposes of test validation, these indices are unsatisfactory because of the different computers and organizational aims.

3.3 The N.I.P.R. Criterion.

3.3.1 Underlying Rationale.

The object in developing a criterion measure for computer programmers was to produce an instrument of performance evaluation which would give a comprehensive, task-anchored account of on-the-job performance in a reliable, yet practicable and economic way.

It was felt that a task-anchored system of assessment would be more objective and bias-free. It was further argued that the rationale underlying a battery of selection tests is based on the premise that these are predictive devices for future on-the-job performance.

3.3.2 Identifying Criterion Schedule Items.

In general, there has been a considerable amount of criticism leveled against supervisory ratings. It has been argued that supervisory assessments are subjective and susceptible to bias and faking. There is, however, no reason to suspect distorted ratings provided that the measuring instrument is sound and the sole purpose of personnel evaluation is test validation. A recent study

has shown that raters in general are astute observers of performance, processing critical on-the-job incidents in a predictable way (21).

The criterion schedule was based on comprehensive job descriptions (20) and other relevant literature (11). A pool of 48 items was established and supervisors had to rate subordinates on a five-point scale. These items dealt with job knowledge, insight and conceptualization the ability to function under pressure, meticulousness and program documentation.

To assess the relevance and comprehensiveness of the preliminary schedule, experts in the computer field were interviewed. To ensure maximum across-industry applicability and to minimize installation or application bound specificities, opinions were sampled over as wide a domain as possible. A large industrial organization, a computer bureau, a mining house, a building society, two supplying firms and a municipality participated.

Every item in the schedule was discussed by the experts, particular attention being paid to its relevance and potential discriminatory ability. In some cases items were rephrased to eliminate terminology specific to a particular type of computer. Some items were eliminated from the schedule because enforced conformity to installation conventions did not permit inter-individual variation.

There was general agreement that the instrument showed promise as a relatively bias-free performance measuring device.

3.3.3 The Criterion Schedule in its final format.

After all the opinions had been considered it was decided to retain 40 items. It was felt that a schedule consisting of 40 items represented a practicable, comprehensive yet manageable instrument. The format selected was a five-point rating scale on each item covering a

wide spectrum from very poor to very good performance. (See Appendix I.) To guard against the associative errors of halo and logical expectation the items were presented randomly. A guide for raters was provided stressing the errors of central tendency, leniency, contrast, halo, and logical expectation. Ratings had to be based on the six month period immediately preceding the assessments to provide equal and sufficiently long periods of observation on each ratee. (See Appendix II.)

3.3.4 Sample of Ratees.

The criterion schedule was administered by supervisors of the Data Processing Services Department of a large industrial organization. Sixty programmers were involved. Table 1 shows mean length and standard deviation of programming experience.

TABLE 1. Mean length and standard deviation of Programming Experience.

Type of Experience	Mean (years) $\bar{x}$	Standard Deviation S.D.	N
Total Experience	2.51	1.34	60
Experience in Organization	2.12	0.81	60

Inspection of the table reveals that the majority of the sample had had little or no outside experience on joining the organization. Follow-up revealed that only 11 of the group of 60 programmers had had previous outside experience ranging between 5 years and 3 months when they first joined the organization.

Formal in-house training was provided as a matter of policy. This training was supplemented by informal on-the-job training so that it may be safely assumed that differences in training did not have a significant effect on ratings. Ratees were all engaged in

programming per se and were not involved in systems analysis.

Of the 60 programmers being assessed, 52 had been selected on aptitude tests on joining the organization. The organization had not adopted a policy of fixed cut-off prints on the tests, but scrutiny of previous test results suggests that considerable preselection on at least intelligence had taken place. (Mean I.Q = 126, S.D. = 7). No information as to the unselected sample of job applicants is, however, available and the possibility exists that the organization attracts superior applicants by virtue of its high standards, good remuneration and fringe benefits.

3.3.5 Reliability of the Criterion Schedule.

The importance of reliability for any measuring device has already been pointed out. The more internally consistent and reliable a measure is, the smaller its standard error of measurement and the more consistently it will rank people on a continuum from poor to good.

The reliability of the criterion schedule was computed by using Kuder-Richardson formula 20 with Ferguson's extension:

$$r_{tt} = \frac{K}{K-1} \quad \frac{\sigma_t^2 - \sum_{i=1}^{k} \sigma_i^2}{\sigma_i^2}$$

where:

K = number of items in measuring device;

σ_t^2 = variance of measuring device;

σ_i^2 = variance of i-th item of device.

Computation yielded a reliability of .96 which is well in excess of the normal requirement of .90 for a sound measuring device.

Table 2 shows the mean, standard deviation, reliability and standard error of measurement of total criterion score.

TABLE 2. Mean, Standard Deviation, Reliability and Standard Error of Measurement for Total Score on Criterion Schedule.

Mean $\bar{x}$	=	140.68
Standard Deviation S.D.	=	25.74
Standard Error of Measurement	=	5.15
Reliability r_{tt}	=	0.96

N = 60

The Standard Error of Measurement was computed by the formula

$$S.E.M. = S.D. \sqrt{1 - r_{tt}} \quad - \quad (1)$$

Substituting in (1), S.E.M. = 5.15.

In practical terms this means that the true score of a person with a score of 130 falls with 95% certainly in the interval 130 $\pm$ 2 S.E.M. = 130 $\pm$ 10.3 i.e. between 119.7 and 140.3.

On the other hand, had the reliability of the instrument been .75, the standard error of measurement would have been 12.87. It could thus be stated with 95% certainty that his true score would have been in the interval 130 $\pm$ 25.74, i.e. between 104.26 and 155.74.

The indices of reliability for each item were also computed according to Gulliksen's method (5). These indices proved to be of satisfactory magnitude.

3.3.6 Relationship between Criterion Score and Programming Experience.

To test the hypothesis that programming proficiency reaches a plateau after a reasonably short time, the correlation between programming experience and criterion score was computed. The correlation was +0.14 which is not significant. In the case of experience within the organization, the correlation approaches zero (0.02) and it may thus be assumed that beyond a certain minimum exposure to a programming environment, capitalization on previous experience is unrelated to proficiency. This statement should, however, be evaluated with reservation because although raters were specifically requested to adopt the same frame of reference irrespective of ratees' previous experience, there is a possibility that leniency towards the less experienced programmers existed.

3.3.7 Relationship between Criterion Score and Merit Ratings.

In the organization where the project was undertaken, annual or sometimes more frequent merit ratings are carried out by using a trait-rating system on all non-managerial personnel. The same system is used consistently, the aim being a universal yard-stick across the organization. The objectives of the system are:

(i) to furnish management with a comprehensive inventory of available manpower resources at any given time;

(ii) to provide information for the award of promotion or special salary increments;

(iii) to identify developmental needs in any particular individual or group of individuals.

The objectives of such a system are thus different from those of test validation where the aim is to predict proficiency in specific tasks. It is clear that

assessments on a merit rating system and a task-specific criterion are neither mutually interchangeable nor strictly comparable. It can, however, be argued that assessments on a job-specific criterion and those on some aspects of a trait rating system should have something in common, both being indices of usefulness to the organization.

Four traits were selected from the merit rating system for comparative purposes, namely Quality of Work, Initiative, Volume of Work and Knowledge of Work. Inspection of their definitions showed that they were essentially job-task oriented and that the underlying rationale was to relate these traits to relevant on-the-job behaviour. (See Appendix III).

In the participating organization, care is exercised in the administration of the merit rating system. Raters are fully conversant with the rationale of the system, ratings are monitored by trained psychologists who attend rating sessions and great emphasis is placed on supporting critical incidents.

Table 3 shows the correlations between the four selected traits and the total criterion score.

TABLE 3. Intercorrelations between Total Criterion Score and Merit Ratings.

Variables Correlated	r_{ij}	
1. Total Criterion Score and Quality of Work	.56	
2. Total Criterion Score and Volume of Work	.45	N=53
3. Total Criterion Score and Initiative	.52	
4. Total Criterion Score and Knowledge of Work	.42	

The above correlations are all significant at the .005 level of confidence showing that these traits are all involved in the total criterion score.

Table 4 shows the intercorrelations between the four traits.

TABLE 4. Intercorrelations between Merit Rating Traits.

Trait	1	2	3	4
1. Quality of Work	1.00	.80	.65	.68
2. Volume of Work		1.00	.69	.70
3. Initiative			1.00	.72
4. Knowledge of Work				1.00

N = 53

The magnitude of the above intercorrelations lends some support to the contention that trait ratings are susceptible to the halo error where raters do not distinguish clearly between traits.

3.3.8 Structure of the Criterion Schedule.

The criterion schedule was designed to measure on-the-job performance by having supervisors rate the performance on a number of activities. In a highly involved and complex job such as programming, the hypothesis may be posed that the criterion is multidimensional.

It is important to determine the underlying constructs clearly and unambiguously if test validation is the main objective, as combining or confusing them will reduce test-criterion correlations.

One of the ways to determine the structure of a set of variables is to subject the intercorrelation matrix of the variables to factor analytic procedures which group variables on the basis of interrelationships.

The intercorrelation matrix was factor analysed and seven groups of items called factors emerged. Interpretation of factors proved to be an extremely difficult exercise in this case. Both orthogonal and

oblique factors were extracted, but neither of these procedures yielded empirically or psychologically meaningful factors. Most items were grouped very closely together, probably as a result of a strong halo tendency. An exception was a factor which clearly indicated documenting proficiency. It should be remembered that the small sample size would also have resulted in large random fluctuations in the correlations.

A third approach was tried. An incidence matrix was formed by assigning weights (according to magnitude) to intercorrelation coefficients, and clusters of items were identified by means of a summation procedure.

The clusters thus identified were plotted from their factor loadings on orthogonal axes. Four meaningful groups of items emerged.

The first item cluster was called General Programming Proficiency. This cluster consisted of items 7, 8, 9, 12, 14, 15, 16, 19, 22, 23, 24, 30, 33, 38 and 40. Close scrutiny of these items revealed an ability to cope with day-to-day programming problems.

Items 5, 13, 20, 21 and 34 related unambiguously to a construct called Documenting Ability.

Careful Methodicalness was characterised by items 1, 2, 4, 7, 18, 22, 25 and 36.

Items 3, 15, 18, 26, 27 and 32 showed a strong element of Programming Logic.

Table 5 shows the intercorrelations between Total Criterion Score and the partial Criteria.

TABLE 5. Intercorrelations between Total Criterion Score and partial Criteria.

Variables	1	2	3	4	5
1. Total Criterion Score	1.00	.95	.45	.89	.86
2. General Programming Proficiency		1.00	.25	.86	.76
3. Documenting Ability			1.00	.26	.50
4. Careful Methodicalness				1.00	.72
5. Programming Logic					1.00

N = 60

It is clear from the above table that Total Criterion Score shares about 90% of its variance with General Programming Proficiency. There is, furthermore, a high relationship between General Programming Proficiency and Careful Methodicalness; a relationship which is not surprising on a priori logical grounds. The relative independence of the construct Documenting Ability is also illustrated.

3.3.9 Relationship between Sub-Criteria and Programming Experience.

To test whether exposure to a programming environment has any effect on performance on the sub-criteria, the correlation coefficients indicated in table 6 were computed.

TABLE 6. Intercorrelations between total length of Programming Experience and Sub-Criteria. N = 60

Variables Correlated	r
Total length of Programming Experience and General Programming Proficiency	.17
Total length of Programming Experience and Documenting Ability	.05
Total Length of Programming Experience and Careful Methodicalness	.12
Total length of Programming Experience and Programming Logic	.24

None of the correlations approach significance so that it may be concluded that proficiency on any of the sub-criteria is statistically unrelated to previous programming experience.

4. TEST VALIDATION.

A procedure for determining test validity concerns itself with the relationship between performance on a test and other independently observable facts about the behaviour under consideration (called the criterion).

4.1 Concurrent and Predictive Validity.

The predictive validity of a test is an index of the effectiveness of a test in predicting a future criterion. This type of information is particularly relevant where decisions pertaining to classification and employment are made (1).

The best strategy is to test a representative sample of the population without making any use of the scores until follow-up results are available. The magnitude of the relationship between test results and criterion data is an index of the predictive validity of the test.

This exercise is, however, seldom possible in practice. Management is essentially pragmatic and insists on the better material being appointed, irrespective of whether the tests have demonstrated validity. Furthermore, it may take several months or even years for criterion data to become available, thereby delaying the implementation of the test battery for an impracticable length of time.

These practical considerations compel the use of concurrent validity i.e. the relationship between simultaneously obtained test and criterion scores. This is not an ideal strategy. It is reasonable to assume that some sort of selective bias operates in any job and that those who 'survive' in a job for a number of years may be substantially different from an unselected sample. It is also difficult to provide adequate

motivation and healthy test-taking attitudes to individuals who are already secure in their jobs.

The need to develop a test battery within a relatively short space of time dictated the use of concurrent validity for this project.

4.2 Validation Sample.

4.2.1 Experience.

The final sample of testees consisted of 49 programmers employed by a large industrial organization. They were a subgroup of the original sample of 60 programmers on which criterion analysis was undertaken. The criterion assessments preceded the testing by approximately 2 months, so that 11 programmers were not available for testing, as they had terminated their services, were on leave or were in hospital.

Table 7 shows the testees' programming experience in total and within the organization.

TABLE 7. Testees' Programming Experience. (N = 49)

	Mean (years)	S.D.
Total length of Experience	2.59	1.45
Length of Experience in Organization	2.11	0.85

The shortest relevant experience was 9 months and the longest 7½ years.

These programmers were fairly homogeneous in terms of hierarchical status and were all designated either Data Processing Assistants Grade I or Data Processing Assistants Grade II.

4.2.2 Previous Selection of Validation Sample.

Forty two of the programmers had been tested on aptitude tests when they joined the organization.

The test battery consisted of a standardized intelligence test, a test of inductive reasoning and a test of verbal reasoning. Later the battery was supplemented by a version of the N.I.P.R. Concept Attainment Test and 27 programmers had taken this test.

Minimum cut-off points were not used on any of these tests, the final decision of employment resting with the Head of Data Processing. The test results are seen as supplementary to other relevant data such as previous experience and motivation. There is, however, full awareness of the potential usefulness of the tests and it is realistic to assume that a positive bias towards test results existed.

4.2.3 Testee attitudes towards Testing.

Testees were expected to undergo the N.I.P.R. testing. There was, however, no evidence of negative or resentful attitudes and co-operation was good. The assignment of code numbers to individuals undoubtedly served to alleviate feelings of threat and of insecurity because anonymity was preserved.

4.3 Relationships between Tests and Criteria.

The job descriptions and their conclusions furnished a broad basis for the formulation of a number of hypotheses regarding the relationship between the cognitive and non-cognitive aspects of the individual on one hand and on-the-job performance on the other hand.

4.3.1 The relationship between Intelligence and Performance.

Findings of various studies cited by Van der Merwe (20) have shown that measures of proficiency and intelligence are correlated. This is not surprising because all jobs consisting of abstract manipulation and conceptualization do require the involvement of intelligence.

The involvement of intelligence is, however, frequently over-estimated leading to over-selection, i.e. the minimum cut-off point on intelligence is set far in excess of the point where a person may still function adequately. This policy is relatively harmless and even beneficial, provided that an adequate selection ratio can be maintained. In a country with a perennial manpower problem, a policy of overselection can only aggravate the problem by artificially creating a vacuum in an already under-manned working sphere.

(a) Hypotheses.

Two alternative hypotheses regarding the involvement of intelligence in routine commercial programming were formulated:

Hypothesis I: There is a positive and statistically significant relationship between intelligence and on-the-job performance;

Hypothesis II: Beyond a certain cut-off, on-the-job performance is independent of intelligence.

(b) Measuring Instrument.

The measuring instrument used was the High Level Mental Alertness Test of the N.I.P.R. It consists of 42 items of the multiple-choice type. Answers are recorded on a separate answer sheet. A time limit of 45 minutes is imposed. It is a power test and approximately 90% of testees finish within the time limit. The items become progressively more difficult and cover a wide field consisting of numerical and letter series, verbal analogies, common elements and other aspects related to reasoning.

(c) Findings.

Table 8 shows the means and standard deviations for the test sample as well as those for a sample of matriculants on the High Level Mental Alertness Test.

TABLE 8. Means and S.D.'s on High Level Mental Alertness Test - Validation Sample and Matriculant Sample.

Sample	N	$\bar{x}$	S.D.
Programmers	49	30	5.3
Matriculants	914	24.9	6.0

Table 9 shows the relationships between the various criteria (including merit ratings) and Mental Alertness.

TABLE 9. Correlations between Mental Alertness, Criteria and Merit Ratings.

Correlation between Mental Alertness and	r_{ij}	
1. Total Criterion Score	-.10	
2. General Programming Proficiency	-.04	
3. Documenting Ability	-.34*	N = 49
4. Careful Methodicalness	-.03	
5. Programming Logic	-.04	
6. Quality of Work	-.09	
7. Volume of Work	-.09	N = 42
8. Knowledge of Work	-.07	

* Significant at 5% level.

Another intelligence test was used for selecting 42 of these programmers when they joined the organization. Table 10 shows the relationship

between the scores on that particular test, the Criteria and the Merit Ratings.

TABLE 10. Intercorrelations between Standardized Intelligence Test, Criteria and Merit Ratings. (N = 42)

Correlation between Standardized Intelligence Test and	r_{ij}
1. Total Criterion Score	-.06
2. General Programming Proficiency	0.00
3. Documenting Ability	-.14
4. Careful Methodicalness	-.21
5. Programming Logic	-.05
6. Quality of Work	-.11
7. Volume of Work	-.12
8. Knowledge of Work	.04

(d) Conclusions.

Table 8 reveals that this particular sample was highly preselected on intelligence. The mean performance of the sample is at the 78th percentile in comparison with a group of matriculants. Further evidence is their previous performance on the standardized intelligence test (Mean 126.1, S.D. 7.5) The mean of a total unselected population is 100 and S.D. = 15.

Table 9 shows that the High Level Mental Alertness Test is not correlated with N.I.P.R., criteria or the merit ratings. The negative correlation between Documenting Ability and High Level Mental Alertness is significant on the 2% level and supports the opinion that the more intelligent programmers do not find the documentation aspect stimulating and tend to neglect it.

Table 10 reveals that the standardized intelligence test used by the organization is also not capable of predicting performance. In practical terms this means that performance on routine commercial programming is independent of intelligence. This statement is, however, strongly qualified by the high degree of preselection which operated on this sample, i.e. this sample functioned on a level of intelligence beyond the optimum level of acceptability and Hypothesis II is accepted.

A word of caution is needed. The natural tendency is to discard tests which show negligible validity. This sample, having survived on the average 2 years of programming as well as an intelligence test on joining the organization is not representative of a sample of job recruits. The intelligence test should be retained to screen those applicants who fall short of the cut-off point.

4.3.2 The Relationship between Temperament and Performance.

It was pointed out by Van der Merwe (20) that programming consists of a set of structured and clearly defined task elements requiring a high degree of perseverance and a flair for detail. Outside social stimulation is minimal although smooth interpersonal relationships are essential where teamwork is concerned.

Every human being can be placed on a continuum of temperament running between the extremes of primary and secondary functioning. An extremely primary functioning person is characterised by a high degree of impulsive behaviour. He reacts quickly to stimuli without evaluating them against a framework of previous experience. He is emotional, susceptible to stimulation

and quick to vent his feelings. He develops strong but momentary enthusiasm for objects or ideals but does not persevere. His moods tend to fluctuate and his level of consciousness is broad but shallow.

An extremely secondary functioning individual is, on the other hand, characterised by a tendency towards meticulousness, slowness and over-cautiousness. He always evaluates his actions against a framework of previous experience and he easily falls prey to fixed emotional patterns. There is strong statistical evidence to suggest that primary functioning is related to extroversion and secondary functioning to introversion.

(a) Hypothesis.

The hypothesis was posed that degree of secondary functioning is positively and significantly related to on-the-job performance.

(b) Measuring Instrument.

The measuring instrument used was the Temperament Questionnaire of the N.I.P.R. The test consists of 27 pairs of short, contrasting descriptions of twofictitious persons called A and B. The testee has to indicate which of A of B more closely resembles himself.

(c) Findings.

Table 11 shows the intercorrelations between various aspects of on-the-job performance and scores on the Temperament Questionnaire. It should be kept in mind that a high score on the test indicates primary functioning and a low score indicates secondary functioning.

TABLE 11. Intercorrelations between Temperament Questionnaire and Criteria and Merit Ratings.

	Temperament Questionnaire and	r_{ij}	N
1.	Total Criterion Score	.12	49
2.	General Programming Proficiency	.16	49
3.	Documenting Ability	-.02	49
4.	Careful Methodicalness	.08	49
5.	Programming Logic	.10	49
6.	Quality of Work	.04	42
7.	Volume of Work	.17	42
8.	Knowledge of Work	-.10	42

(d) Conclusions.

None of the above correlations approaches significance.

There is no statistical evidence to accept the hypothesis and it may be concluded that, for this sample, temperament is irrelevant in terms of on-the-job performance.

4.3.3 The relationship between problem solving ability and on-the-job Performance.

It was suggested by Van der Merwe (20) that the selection procedures should aim at simulating the problem-solving aspect of programming i.e. "require the structuring of a number of steps to achieve a given objective and the reformulating of this structure in terms of very specific restrictions" (p. 59).

(a) Hypothesis.

The hypothesis was formulated that a positive and statistically significant relationship exists between the ability to solve problems under restrictions and on-the-job performance.

(b) Measuring Instrument.

(i) Rationale.

This test, called Concept Attainment Form B, was developed by Schepers in 1958. The major considerations taken into account at the inception and development of the test were:

(a) It had to be suitable for group testing;

(b) It had to conform to the metric requirement of reliability, i.e. it had to consist of more than one item of the same kind;

(c) The testee had to leave a permanent record of what he had done so that scoring could be done away from the test room.

The result was a test which had certain similarities to an individual test developed by Maag (9). It was, however, essentially dissimilar in the sense that Schepers's test was primarily aimed at eliciting a specific, rational strategy by placing definite restrictions on the testee.

The version used in this study consists of 32 line drawings of cubical objects consecutively numbered from 1 to 32. No two objects are exactly alike and there are five ways in which they can differ from one another. Sixteen of the objects have any one of these characteristics in common, eight have a combination of any two, etc.

The testee is confronted with a particular object and told that this object has two characteristics in common with seven other objects. His task is to identify these two characteristics and to indicate the seven other objects sharing those two characteristics.

The testee has to identify the common characteristics by selecting any other object. After each choice he is informed whether he had made a 'True' or a 'False' choice. Four choices are permitted - the maximum permissable number to yield full information and to lead to the elimination of 9 hypotheses, provided that a rational strategy has been adopted.

Permanent record of problem solving behaviour is provided by the design of the answer sheet.

The test consists of 10 items. The first five problems can be solved by adopting a 'single change' strategy i.e. by varying one characteristic at a time but the second five problems can only be solved by a 'double change' strategy i.e. by varying two characteristics simultaneously. The time limit is 70 minutes.

(ii) Scoring Components of the Test.*

The test yields various components which furnish information as to the problem solving ability of the testee and which avail themselves to objective scoring. They are the following:

(a) The total number of exemplars correctly identified;

(b) the total number of problems for which the testee manages to elicit full information - i.e. for which he eliminates 9 hypotheses;

(c) the total number of items in the first five problems for which a rational single-change strategy is followed;

(d) a total strategy score of the last 5 items in which the testee is compelled to follow a double-change strategy. Each such double change strategy is assigned a weight of 2;

* A program developed by the N.I.P.R. for an I.B.M. 360 type 50 computer was used to score the test.

(e) a total strategy score consisting of the sum of (c) and (d).

Table 12 indicates the intercorrelations between the different scores.

TABLE 12. Intercorrelations between different Scores of the Concept Attainment Test.

Variables	1	2	3	4	5
1. Total Number of Exemplars	1.00	.92	.57	.70	.72
2. Number of items with complete information		1.00	.71	.71	.79
3. Single-change strategy scores			1.00	.60	.87
4. Double-change strategy scores				1.00	.92
5. Total Strategy Score					1.00

The high correlations between some indices are to be expected by virtue of the high degree of experimental dependence, some intercorrelations being part-whole. The relatively low intercorrelation between variables 1 and 3 can be explained by the fact that these variables measure relatively different qualities. Variable 1 is an index not only of the rationality of choices and strategy followed, but also of the ability to integrate information. (Some testees follow a gamble strategy and yet manage to elicit full information.) Variable 3 is a measure of the rationality and acceptability of the strategy pursued in the first five problems only, so that there is justification for including both variables as relatively independent entities.

(iii) Relationship between Intelligence and Problem Solving Ability.

Table 13 shows the correlations between Mental Alertness and the components of the Concept Attainment Test.

TABLE 13. Correlations between Concept Attainment and High Level Mental Alertness. (N = 49)

High Level Mental Alertness and	r_{ij}
1. Total Number of Exemplars Identified	-.01
2. Number of Items with complete Information	-.03
3. Sum of Single-change Strategy Scores	.16
4. Sum of Double-change Strategy Scores	.21
5. Total Strategy Score	.21

There is no statistical evidence to suggest that intelligence, as measured by the Mental Alertness Test, is significantly involved in Problem Solving Ability as measured by the Concept Attainment Test. Earlier work by Schepers (12) had also demonstrated that the Concept Attainment Test sampled a relatively unique cognitive domain.

(c) Findings.

(i) Relationships between N.I.P.R. Criteria and Concept Attainment.

The intercorrelations between the components of the N.I.P.R. Criterion Schedule and the various measures of the Concept Attainment Test are presented in Table 14.

TABLE 14. Intercorrelations - N.I.P.R. Criteria and Components of Concept Attainment Test. (N = 49)

Variables	Number of Exemplars	Items with full Inform.	Single-change Strats.	Double-change Strats.	Total Strategy Score
1. Total Criterion Score	.37**	.33*	.41**	.17	.31*
2. General Programming Proficiency	.31*	.29*	.36**	.18	.29*
3. Documenting Ability	.26	.18	.02	-.14	-.08
4. Careful Methodicalness	.32*	.29*	.39**	.17	.30*
5. Programming Logic	.45**	.45**	.50**	.24	.40**

* Denotes significance at 5% level of confidence.

** Denotes significance at 1% level of confidence.

Number of exemplars correctly identified and Sum of Single-change strategies are relatively independent measures as indicated before and the multiple correlations between these entities and the N.I.P.R. criteria were computed according to the formula (6)

$$R_{i.jk}^2 = \frac{r_{ij}^2 + r_{ik}^2 - 2r_{ij}\, r_{ik}\, r_{jk}}{1 - r_{jk}^2}$$

where $R_{i.jk}$ is the multiple correlation coefficient between variable i and variables j and k;

r_{ij} is the correlation between variable i and variable j;

r_{ik} is the correlation between variable i and variable k;

r_{jk} is the correlation between variable j and variable k.

Table 15 shows the multiple correlations between N.I.P.R. criteria and Number of Exemplars correctly identified and Sum of Single-change Strategies.

TABLE 15. Number of Exemplars correctly Identified and Sum of Single-change Strategies.

Variables Correlated	$R_{i.jk}$
1. Total Criterion Score and (Number of Exemplars and Sum of Single Strategy Scores)	.44
2. General Programming Proficiency (Number of Exemplars and Sum of Single Strategy Scores)	.38
3. Careful Methodicalness (Number of Exemplars and Sum of Single Strategy Scores)	.41
4. Programming Logic	.54

The probability that correlations of these magnitudes could have arisen by random variation is smaller than 0.1%.

(ii) Relationship between Merit Ratings and Concept Attainment.

Table 16 shows the intercorrelations between Merit Ratings and the components of the Concept Attainment Test.

TABLE 16. Correlations between Merit Ratings and Components of the Concept Attainment Test. (N = 42)

Variables	Exemplars Identified	Items with full Inform.	Sum of Single-Strats.	Sum of Double-Strats.	Total Strat. Score
Quality of work	.37*	.35*	.36*	.27	.36*
Volume of work	.25	.24	.28	.26	.31*
Knowledge of work	.32*	.31*	.24	.28	.31*

* Significant at 5% level of confidence.

(d) Conclusions.

Inspection of Table 14 shows that all the components of the Concept Attainment Test (with the exception of Sum of double-change Strategies) relate significantly to Total Criterion Score, General Programming Proficiency, Careful Methodicalness and Programming Logic. Documentation Ability is not predicted by any of these components. This is hardly surprising, the abilities needed for documentation diverging widely from the abilities being measured.

Sum of single-change Strategies is consistently the best predictor which shows that the testee who develops a rational and consistent approach in this test transfers this ability to the working situation.

Sum of double-change Strategies does not show any appreciable validity, the reasons probably being the high difficulty value of the last five items plus the fact that an additional source of variance is introduced through testees not all completing the last five items.

Total number of exemplars identified has a strong element of integration ability. Validity coefficients show the involvement of this ability in programming where bits of information have to be integrated and accommodated in a broad conceptual framework.

Table 15 shows that prediction can be improved by mathematically combining Total Number of Exemplars identified and Sum of single-change Strategies.

Table 16 shows that three selected traits from the Merit Ratings are consistently related

to Total Strategy Score. Volume of Work is most highly associated with Total Number of Exemplars identified. The same holds for Knowledge of Work.

In general, it may be concluded that Conceptual reasoning and its various components are highly involved in programming proficiency, showing a statistically significant relationship with a wide diversity of on-the-job criteria. The hypothesis is thus accepted.

4.3.4 The Relationship between the Ability to Integrate Information and on-the-job Performance.

Programming does not only consist of the systematic search for and conceptualization of information, but also of the translation into symbolic language of such information subject to certain specific restrictions. The integration of information into a meaningful system of unambiguous symbols forms an important part of a programmer's task.

(a) Hypothesis.

The hypothesis was posed that a positive and statistically significant relationship exists between the ability to integrate information and on-the-job proficiency.

(b) Measuring Instrument.

The test used to measure this ability is called the Concept Attainment Test Form B (Part II). It differs from the Concept Attainment Test previously described in that in this test, the testee is confronted with sufficient feedback about a particular object to enable him to integrate this information in such a way that he can arrive at a solution.

In essence this means that the testee is not required to develop a strategy, but that his task is utilize information in such a way as to lead to the identification of seven exemplars which share two characteristics with a given model.

There are 10 problems and the time limit is 60 minutes.

(c) Findings.

Table 17 shows the mean and standard deviation of test scores on this particular test.

TABLE 17. Mean and S.D. for the Concept Attainment Test Form B (Part II).

Mean $\bar{x}$	59.6
Standard Deviation S.D.	10.8
N	49

Table 18 shows the intercorrelation between Concept Attainment Form B (Part II) and Criterion Measures.

TABLE 18. Correlations between the Concept Attainment Form B (Part II) and Criteria (including Merit Ratings).

Correlations of Concept Attainment Form B (Part II) with	r_{ij}	N
1. Total Criterion Score	.14	49
2. General Programming Proficiency	.13	49
3. Documenting Ability	-.01	49
4. Careful Methodicalness	.13	49
5. Programming Logic	.13	49
6. Quality of Work	.17	42
7. Volume of Work	.18	42
8. Job Knowledge	.13	42

(d) Conclusions.

The high mean performance of the test shows that it was too easy and therefore did not measure individual differences reliably - inspection of the raw scores revealed that 12 testees gained full marks on the test. The reason is probably the fact that the first part of the test had immediately preceded this particular test so that transfer of knowledge occurred. Test sophistication, by virtue of the high degree of similarity between the tests, cannot be ruled out.

The correlations reported in Table 18 do not support the hypothesis. The rejection of the hypothesis should, however, be seen in the experimental context described.

5. CONCLUSIONS AND RECOMMENDATIONS.

This pilot study provided sufficient information to serve as a basis for further follow-up studies and research.

5.1 Criterion Refinement.

The Criterion Schedule is highly satisfactory in terms of its discriminatory ability. Further refinement is, however, called for.

It is felt that basic constructs such as Programming Logic, Programming Proficiency, Insight into the Functions Programs have to Perform, Careful Methodicalness should be properly defined and items generated which refer specifically to these *a priori* constructs. It is further recommended that a sufficient number of items per empirical construct be generated so that the reliability of each construct is optimal.

5.2 The Involvement of Intelligence in Programming.

Further research into the rôle played by intelligence is indicated to ensure the determination of a realistic cut-off point. It is felt that an over-emphasis on the involvement of

intelligence may lead to over-selection and the aggravation of an already serious manpower problem. A realistic cut-off point would ensure a larger influx of potentially useful material into programming without compromising proficiency.

5.3 The Involvement of Problem Solving Ability.

The findings regarding the involvement of this ability are highly encouraging. The relationship between problem solving ability and on-the-job proficiency is probably higher than indicated by the correlation coefficients if the assumption that the sample is preselected, is tenable. A considerable proportion of the sample was re-tested on the Concept Attainment Test and the effects of test sophistication could also have resulted in diminished validity coefficients.

The second half of this particular version of the Concept Attainment Test was too difficult and it is recommended that this part of the test be replaced by a parallel version of the first half which showed the highest validity. In this way the reliability of the best predictor could be considerably enhanced.

5.4 Unresolved Aspects.

It is clear that a number of problems are unresolved and two further studies are planned to clarify these aspects.

5.4.1 The Involvement of Reasoning Ability.

The involvement of 'reasoning ability' though not properly defined, has been strongly emphasized by McNamara and Hughes (8) who based their Programmer Aptitude Test and its revised version on this assumption. Coefficients of validity point strongly to a high degree of involvement of this specific ability (7, 8). Although these findings were not supported by data gathered in a nation-wide survey in the United States and Canada (8), it could be argued that criterion

measures could have been unsatisfactory in many cases.

The job descriptions (20) and the findings of McNamara and McNamara and of Hughes cited above present sufficient evidence for the formulation of the following hypothesis:

There is a positive and statistically significant relationship between reasoning ability, as measured by the Gottschaldt Figures Test and the N.I.P.R. Pattern Relations Test and measures of on-the-job proficiency.

5.4.2 The Involvement of Career Interest Patterns.

The suitability of any candidate for a particular job is not only determined by his unique configuration of abilities, but is also strongly influenced by his motivation.

Van der Merwe (20) summarized the work of Perry and Cannon as follows: "The striking characteristic of programmers is their interest in problem- and puzzle solving activities, and a combination of applied scientific and administrative interests involving technological application rather than theory. Generally speaking, computer programmers are different from other professional men in their greater interest in problem solving, mathematics and mechanical pursuits and their lesser interest in people, especially activities involving personal interaction" (p. 14).

The following hypothesis, based on psychological grounds as well as the above findings, is presented:

Successful and unsuccessful programmers display differential career interest patterns.

5.4.3 The Involvement of Speed and Accuracy.

In a commercial programming environment, speed and accuracy in detail emerge as desirable attributes. These have not been adequately explored by previous investigators, but it is felt that the strong emphasis on these two qualities in the job situation warrants the testing of the following hypothesis:

Clerical speed and accuracy are both positively and significantly related to on-the-job proficiency.

REFERENCES

1. Anastasi, A.: Psychological Testing (Second Edition) New York, The Macmillan Company, 1961.

2. Berger, R.M. and R.C. Wilson: The development of programmer evaluation Measures. In: Proceedings of the third Annual Computer Personnel Research Conference, June 17 - 18, 1965, pp 6 - 17.

3. Berger, R.M. and R.C. Wilson: Correlates of Programmer Proficiency. In: Proceedings of the fourth Annual Computer Personnel Research Conference, June 27 - 28, 1966 pp. 83 - 85.

4. Brown, C.W. and E.E. Ghiselli.: The relationship between predictive power of aptitude tests for trainability and for job proficiency. J. appl. Psychol., 1952, 36, 370 - 372.

5. Gulliksen, H.: Theory of Mental Tests, New York, John Wiley, 1950.

6. Lord, F.M. and M.R. Novick: Statistical theories of mental test scores. Reading, Mass. Addison Wesley Publishing Company, 1968.

7. McNamara, W.J.: The selection of Computer Personnel-Past, Present and Future. In: Proceedings of the fifth Annual Computer Personnel Research Conference. June 26 - 27, 1967, pp 52 - 56.

8. McNamara, W.J. and J.L. Hughes: A review of the selection of Computer Programmers. Personn. Psychol., 1961, 14, 39 - 51.

9. Maag, C.H.: Development and evaluation of a Conceptual Reasoning Test. Educ. and Psychol. Measmt. 1957, 17, 230 - 239.

10. Nagle, B.F.: Criterion Development. Personn. Psychol. 1953, 6, 271 - 289.

11. Rigney, J.W., R.M. Berger and A. Gersohn: Computer Research Selection and Criterion Development : I The Research Plans. Washington D.C., Department of Commerce, 1963 (University of Southern California, Department of Psychology Project, NR 153 - 093, Technical Report No. 36.)

12. Schepers, J.M.: Unpublished Report. Johannesburg, National Institute for Personnel Research, Johannesburg, 1958.

13. Schultz, D.G. and A.I. Siegel: Progress and problems in the measurement of individual differences in on-the-job performance. Acta psychol., 1963, 21, 120 - 156.

14. Severin, D.G.: The predictability of various kinds of criteria. Personn. Psychol., 1952, 5, 93 - 104.

15. Spector, A.J.: Influences on merit ratings. J. appl. Psychol., 1954, 38, 393 - 396.

16. Stalnaker, A.W.: The Watson-Glaser Critical Thinking Appraisal as a predictor of programming performance. In: Proceedings of the third Annual Computer Personnel Research Conference, June 17 - 18, 1965, pp. 75 - 77.

17. Sydiaha, D.: Criterion reliability: a study of the folkways and mores of the psychological profession. The Canadian Psychologist, 1964, 5a, 17 - 33.

18. Taylor, E.K. and E.G. Nevis: Personnel Selection. A Rev. Psychol., 1961, 12, 389 - 412.

19. Turner, W.W.: Dimensions of foreman performance: a factor analysis of criterion measures. J. appl. Psychol., 1960, 44, 216 - 223.

20. Van der Merwe, Margaretha: Operator, Programmer and System Analyst Job Demands: A description and analysis. Report submitted to The Computer Society of South Africa, Johannesburg. National Institute for Personnel Research (C.S.I.R.) 1968.

21. Whitlock, G.H.: Application of the psychophysical law to performance evaluation. J. appl. Psychol., 1963, 47, 15 - 23.

APPENDIX I

NATIONAL INSTITUTE FOR PERSONNEL RESEARCH

Computer Programmer Evaluation Schedule

CONFIDENTIAL: This material must not be shown to unauthorized persons or used without permission of the National Institute for Personnel Research.

Ratee:

Name: .....

Designation: .....

Programming Experience in current organization ..... years .....months

Programming Experience in previous organization(s) ..... years .....months

Rater:

Name: .....

Designation: .....

Programming experience ..... years ..... months

Date: .....

1. Does he use testing techniques wastefully?

1	2	3	4	5
Very frequently				Very seldom

2. How many compiling or assembler runs does he need to remove syntax errors from his programs?

1	2	3	4	5
A large number				A small number

3. Do his programs require excessively long running times?

1	2	3	4	5
Very frequently				Very seldom

4. How often, after he has made apparently minor changes to a program, has the number of test runs shown that he has not considered the implications carefully enough?

1	2	3	4	5
Very frequently				Very seldom

5. Are his operating instructions referred back to him because of obscurities?

1	2	3	4	5
Very frequently				Very seldom

6. How often has he not understood the logic of a program before attempting to write it?

1	2	3	4	5
Very frequently				Very seldom

7. Do errors, that should have been brought out by his test data, appear during runs of his programs?

1	2	3	4	5
Very frequently				Very seldom

8. Is he quick at grasping the program specifications and all its implications?

1	2	3	4	5
Very slow				Very quick

9. Do you give him programs that must meet tight deadlines?

1	2	3	4	5
Very seldom				Very frequently

10. Does he have to have the statements in technical manuals explained to him?

1	2	3	4	5
Very frequently				Very seldom

11. Does he ask about the causes of common error messages?

1	2	3	4	5
Very frequently				Very seldom

12. How good is his knowledge of coding techniques?

1	2	3	4	5
Very poor				Very good

13. How often does his documentation not conform to installation standards (e.g. adherence to prescribed headings, etc.)?

1	2	3	4	5
Very frequently				Very seldom

14. If you had a problem in a program, would you refer it to him?

1	2	3	4	5
Very seldom				Very frequently

15. Do you have to point out to him how he can improve the running time of his program by modifying program logic?

1	2	3	4	5
Very frequently				Very seldom

16. Before writing a program, does he notice and point out important information that has been overlooked?

1	2	3	4	5
Very seldom				Very frequently

17. How many test runs does he need to get a program of average length and complexity working properly?

1	2	3	4	5
Well above average				Well below average

18. Does he have unnecessary trouble in sorting out programming errors due to the complexity of his logic?

1	2	3	4	5
Very frequently				Very seldom

19. How good is his knowledge of <u>new</u> programming techniques?

1	2	3	4	5
Very poor				Very good

20. Do you have to point out to him that his documentation is poor because he has put in too little information?

1	2	3	4	5
Very frequently				Very seldom

21. Does he not keep to installation conventions without valid reason?

1	2	3	4	5
Very frequently				Very seldom

22. Would you under critical circumstances ask him to adapt a program written by another programmer?

1	2	3	4	5
Very seldom				Very frequently

23. How often would you hand him a program that requires sophisticated programming techniques?

1	2	3	4	5
Very seldom				Very frequently

24. Does he notice apparent discrepancies or contradictions in program specifications given to him?

1	2	3	4	5
Very seldom				Very frequently

25. Rate the accuracy of his coding.

1	2	3	4	5
Very inaccurate				Very accurate

26. Does he use correct but obscure logic in his flow charts?

1	2	3	4	5
Very frequently				Very seldom

27. Does he use subroutines or segment his programs where it would be advantageous for him to do so?

1	2	3	4	5
Very seldom				Very frequently

28. How frequently would you give him basic flow charts to expand into program flow charts?

1	2	3	4	5
Very seldom				Very frequently

29. Is he inclined to include too much detail in his documentation?

1	2	3	4	5
Very frequently				Very seldom

30. Does it take him long to utilize new programming facilities?

1	2	3	4	5
Very long time				Very short time

31. Does he use less commonly known statements or instructions in his programs?

1	2	3	4	5
Very seldom				Very frequently

32. Are his programs needlessly complex?

1	2	3	4	5
Very often				Very seldom

33. Is he flexible in adapting his programming methods to varying requirements?

1	2	3	4	5
Very rigid				Very flexible

34. Does he use comments and meaningful names to make his programs more intelligible?

1	2	3	4	5
Very seldom				Very frequently

35. How many assignments is he capable of handling simultaneously?

1	2	3	4	5
A very limited number				A very large number

36. Does he think out the logic of a program properly before starting to write it?

1	2	3	4	5
Very seldom				Consistently

37. Is his productivity impaired by working under pressure?

1	2	3	4	5
To a large extent				To a small extent

38. Does he recognise the need for new programs or subroutines on his own initiative?

1	2	3	4	5
Very seldom				Very frequently

39. How does he cope with on-the-job frustrations?

1	2	3	4	5
Very inadequately				Very adequately

40. How frequently would you use him for undertaking on-the-job training of other programmers?

1	2	3	4	5
Very seldom				Very frequently

APPENDIX II

GUIDE TO COMPUTER PROGRAMMER EVALUATION SCHEDULE

These ratings must be made on actual on-the-job performance over the past six months. No assessments pertaining to personality traits or character attributes are required.

Your task is to rate each employee on the questions in the schedule. Circle the number on the five-point scale which in your opinion is a fair representation of his performance over the past six months.

If any particular question is not relevant for the ratee, mark it N/A.

The validity of your assessments is greatly enhanced by following a few simple rules:

1. Avoid the 'halo effect'. This is a tendency to rate a particular individual about the same on all aspects of his performance because of a general impression whether favourable or unfavourable. Try to see each element of the individual's performance independently.

2. Avoid the leniency error. A lenient rater rates almost every ratee as competent and satisfactory. He avoids unfavourable assessments because he does not want to hurt feelings. A rater who gives a preponderance of negative ratings is equally at fault. A good rater avoids both positive and negative leniency and gives balanced ratings.

3. Utilize both extremes of the scale. Some raters do not use points 1 and 5 on a five-point scale and tend to group ratings around the average which reduces the scale to a three-point scale.

4. Do not assume any relationship between various elements of performance. This error, unlike the 'halo' which occurs when the apparent coherence of qualities in the same individual is assumed, occurs when a rater assumes relationships between elements of performance irrespective of individuals.

5. Avoid the contrast error. This is a tendency for a rater to rate others in the opposite direction from himself on certain items. A rater who excels in certain activities tends to see all others as inferior to himself.

A rough interpretation of the scales is 1 or 5 extreme, 2 or 4 above or below average, 3 average.

Some of the questions are very similar in content. This is intentional and was built into the instrument for future statistical analysis. Respond to each statement independently and please do not refer back to previous statements. Thank you for your co-operation.

APPENDIX III

SELECTED TRAITS FROM MERIT RATING SYSTEM

QUALITY OF WORK

To what extent is his work free from mistakes? Consider accuracy, thoroughness, neatness and usefulness of the final product.

1	2	3	4	5	6	7	8	9
Consistently unsatisfactory.		Unsatisfactory at times.		Maintains a good quality.		Renders a very good all-round quality.		Consistently maintains an exceptionally high standard.

VOLUME OF WORK

At what tempo does he work, and what is his productivity?

1	2	3	4	5	6	7	8	9
Extremely slow worker.		Experiences difficulty in meeting schedules.		Maintains a good and consistent output.		High tempo of work. Often handles additional work loads.		Maintains an exceptionally high output.

INITIATIVE

To what extent does he contribute original and constructive ideas? How does he act in unfamiliar or unforeseen situations?

1	2	3	4	5	6	7	8	9
Never contributes original and workable ideas		Sometimes produces ideas of minor importance.		Contributes good and workable ideas.		Shows considerable ability to produce new and practical ideas. Results are significant.		Most ingenious and skilful, also with regard to future possibilities.

KNOWLEDGE OF JOB

How is his knowledge of his job? To what extent is he dependent upon guidance and additional information?

1	2	3	4	5	6	7	8	9
Receives guidance, also in regard to routine tasks.		Fair knowledge. Receives guidance and additional information		Possesses a good knowledge. Requires no guidance in routine tasks.		Possesses a wide and extensive knowledge.		Has an exceptionally thorough knowledge of his job and all related matters.