1985

Work-Related Correlates of Job Satisfaction in Programmer/Analysts: An Examination of Task Differences

David K. Goldstein
Harvard Business School

Work-Related Correlates of Job Satisfaction in Programmer/Analysts: An Examination of Task Differences

David K. Goldstein
Harvard Business School
Boston, Massachusetts
May, 1985

ABSTRACT

The present research examined task differences in work satisfaction and in several correlates of satisfaction among programmer/analysts. Subjects were initially classified along analysis/programming and development/maintenance dimensions. Questionnaire data were collected from over 250 subjects at a large manufacturing company. The results indicated that analysts were more satisfied with their work than programmers and that developers were more satisfied than maintainers. In a secondary analysis, subjects were divided into five groups: development/programmers, maintenance/programmers, development/analysts, maintenance/analysts and supporters—those who primarily provided user and production support. The results indicated that maintenance/programmers were less satisfied than the other four groups—who were approximately equally satisfied. Significant differences in skill variety, autonomy, role ambiguity and amount of user interaction also existed among the five groups. A revised task differences model was presented based on this secondary analysis.

Introduction

With organizations increasing their use of computer-based information systems, the work carried out by programmer/analysts—those who develop and maintain these systems—has become increasingly critical. As such, the extent to which these individuals are satisfied with their work has become a rising concern among information systems managers. The present research focuses on one aspect of job satisfaction among these individuals—the relationship between the tasks performed by programmer/analysts and the degree to which they are satisfied with their work.

With the growth of information systems staffs, the job performed by these staffs has become more specialized. For example, the large number of installed information systems has led some companies to assign programmer/analysts to the sole activity of maintaining these systems. If working on different tasks leads to differences in satisfaction among programmer/analysts, we cannot consider these workers as one group. We must develop separate strategies for improving the work for the different groups of programmer/analysts. In addition, we must separately examine the impact of technological changes on the different groups.

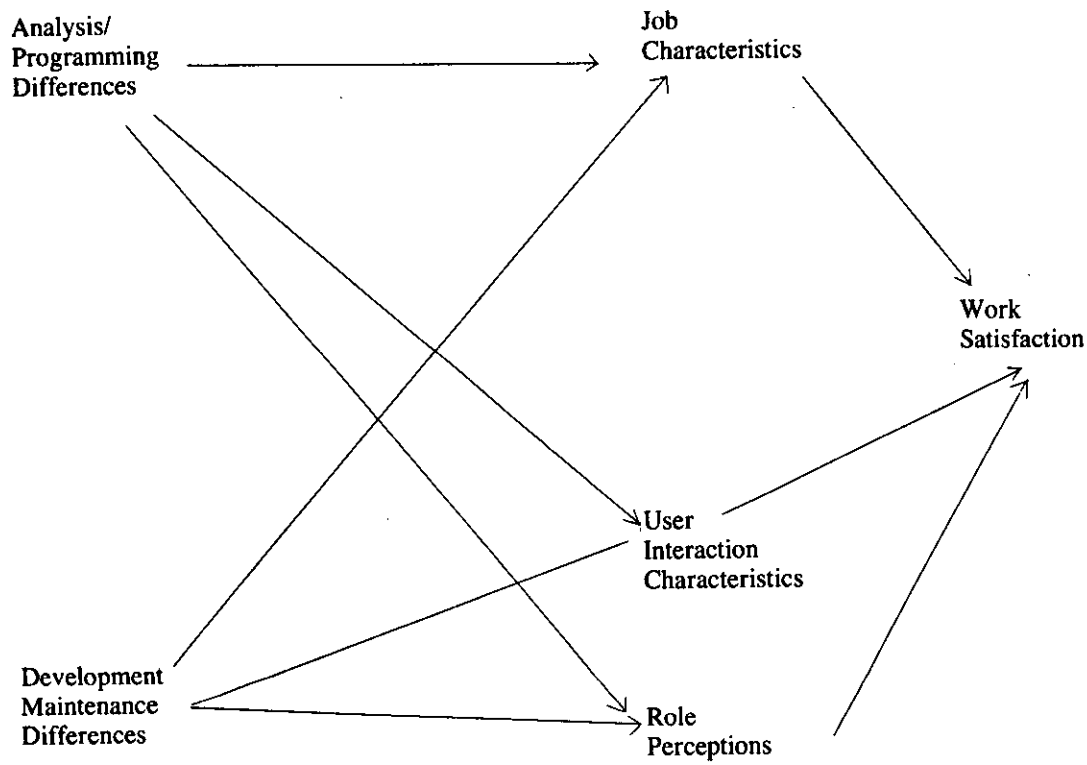
There are two objectives to this study. The first is to determine if there are differences in satisfaction between programmer/analysts that can be attributed to differences in tasks. Programmer/analysts are classified along two task dimensions—into developers and maintainers and into analysts and programmers. The second is to examine the extent to which task differences in satisfaction can be explained by three sets of variables—job characteristics, role perceptions, and user interaction characteristics (Figure 1). These variables have been shown to be determinants of satisfaction in previous studies (Couger and Zawacki, 1981; Goldstein and Rockart, 1984).

Several researchers have examined the effect of task differences on job satisfaction or on related variables. Couger and Zawacki (1981) considered differences between analysts, programmers, and programmer/analysts using job titles as a means of classifying subjects. They found programmers were more satisfied than analysts, even though analysts scored more highly on each of the five job characteristics. Baroudi (1984) found positive correlations between the amount of analytic duties performed by a programmer/analyst and three variables—boundary spanning, role conflict, and organizational commitment. He found negative correlations between analytic duties and role ambiguity.

In separate papers, these researchers also examined differences between maintainers and developers. Couger and Cotler (1984) found a negative correlation between the percentage of time spent maintaining systems and both motivating potential score and role conflict. Baroudi and Ginzberg (1984) found a negative correlation between percentage of time spent on maintenance and satisfaction. They found no significant correlation between

Figure 1

Proposed Model of the Impact of Task Differences on Work Satisfaction



maintenance percentage and either organizational commitment, role ambiguity, or role conflict.

This research builds on the studies discussed above. It considers both analysis/programming and maintenance/development differences. This approach allows us to determine if there are interactions between these task differences variables. The research employs multiple measures of task differences permitting us to assess the measures' reliability. In addition, it considers a larger set of explanatory variables than were considered in the above studies.

Task Differences Hypotheses

In this section, the hypothesized differences between maintainers and developers and between analysts and programmers are presented. There are three parts to this section. In the first subsection, the variables used in this study are described. In the last two subsections, differences between maintainers and developers and between analysts and programmers are discussed.

VARIABLES TO BE STUDIED

Three sets of variables are included in this study. The first set is the task differences variables. They are used to classify programmer/analysts as either developers or maintainers and as either analysts or programmers. The second set of variables consists of the dependent variable—work satisfaction.

The third set consists of three groups of explanatory variables—variables that are hypothesized to be affected by task differences and that, in turn, should affect work satisfaction. The choice of explanatory variables is based on the model presented in Goldstein (1985).

The Job Characteristics Model (JCM) (Hackman and Oldham, 1976, 1980) provides the first set of explanatory

variables. This model specifies that five objective characteristics of work—skill variety, task identity, task significance, autonomy, and feedback from the job—are related to several outcome variables, including satisfaction. Goldstein and Rockart (1984) and Couger and Zawacki (1980) found significant correlations between job characteristics and satisfaction in studies of programmer/analysts.

The second set are characteristics of interactions between programmer/analysts and users. This author (Goldstein, 1985) identified two variables that measure the quality and quantity of user-programmer/analyst interactions. They are:

> *user feedback*—the degree to which the users of the system on which the programmer/analyst is working provide the programmer/analyst with direct and clear information about the effectiveness of his or her performance.
>
> *user interaction*—the degree to which the programmer/analyst's job requires him or her to work closely with users in carrying out the work activities.

Both these variables are hypothesized to be positively related to work satisfaction in programmer/analysts.

The third group of variables are role perceptions. They measure the amount of conflict and ambiguity perceived by programmer/analysts. Goldstein and Rockart (1984) and Bostrom (1980) found a negative correlation between work satisfaction and both role conflict and ambiguity.

DIFFERENCES BETWEEN MAINTAINERS AND DEVELOPERS

There are several characteristics that distinguish the work carried out by developers from the work carried out by maintainers. First, there are several functions which are unique to developers, such as systems design, and other functions which are unique to maintainers, such as tracing through the logic of an existing program. Second, developers work on projects that take between six months and several years to complete (Walston and Felix, 1977), while maintainers work on projects that last for several days or for weeks (Fjelstad and Hamlen, 1977). Third, developers are likely to work in teams, while maintainers are likely to work one-to-one with users. Fourth, the development process consists of well-established steps with specific milestones and deliverables. The maintenance process is more ad hoc.

These differences should affect both the explanatory variables and work satisfaction. Hypotheses predicting these differences are outlined below.

H1: Skill variety will be higher for developers. A wider variety of skills—including those needed to determine systems requirements and to design systems—is needed in the development of information systems than in their maintenance.

H2: Task significance will be higher for developers. The development of a new system has more potential for making a major change in the way a company operates than the change to an existing system.

H3: Autonomy will be higher for maintainers. Since maintainers work more indepentently than developers, they should have more freedom in planning and scheduling their work.

H4: Feedback from the job should be higher for maintainers.

H5: User feedback will be higher for maintainers. The shorter duration of the maintenance projects should provide maintainers with more opportunity for both user and job feedback than developers.

H6: User interaction will be higher for maintainers. Developers have little interaction with users during system design and programming, while maintainers must interact with users thoughout the maintenance process.

H7: Role ambiguity will be lower in developers. The software development lifecycle provides developers with a clear set of guideline, reducing the amount of ambiguity that they perceive.

H8: Role conflict will be lower for developers. Maintainers should experience higher levels of role conflict because, in many cases, they may not have a clear idea of why they are making a specific change.

H9: Work satisfaction should be higher for developers. Lower levels of role ambiguity and role conflict and higher levels of skill variety and task significance should have a strong positive effect on the work satisfaction of developers. It should outweigh the negative effect of lower autonomy, job feedback, user feedback, and user interaction. The general feeling among programmer/analysts that development is more presitgious than maintenance could also have a positive influence on the work satisfaction of developers.

Differences Between Analysts and Programmers

The differences between programming and analysis lie mainly in differences in the functions performed in carrying out the two tasks. There are several functions unique to analysts including determining system requirements

(in development) or determining change requirements (in maintenance). Programming, whether in development or maintenance, involves writing programs and testing to see if the programs work. The differences between analysts and programmers are discussed in the hypothesis below.

H10: Skill variety will be higher for analysts. The number and complexity of skills needed to carry out systems analysis and design are much greater than those needed to write or modify programs.

H11: Task identity will be higher for programmers. Since the development of a module or the modification of a program is a whole piece of work, programmers will experience a higher level of task identity.

H12: Task significance will be higher for analysts. The systems analysis and design of an information system should make a more significant contribution to the organization than the writing or modification of a program.

H13: Autonomy will be higher for analysts. Programmers are more limited in the amount of autonomy they have, because their work is often planned and scheduled for them by analysts or by their supervisors.

H14: Feedback from the job will be higher in programmers. The shorter duration of the tasks performed by programmers will lead to more feedback.

H15: User Feedback will be higher for analysts.

H16: User interaction will be higher for analysts. Programmers interact with and receive feedback from their supervisor or an analyst in most cases, while analysts are more likely to interact with and receive from users.

H17: Role ambiguity will be lower in programmers. The analysis and design phases are less structured than the programming phase both in systems development and in maintenance.

H18: Role conflict will be lower for analysts. Programmers face high levels of role conflict, because the tasks they perform are often dictated by others (users or their supervisors).

H19: Work satisfaction will be higher for analysts. The higher levels of skill variety and autonomy combined with lower role conflict will lead analysts to experience more satisfaction with their work than programmers. This will be somewhat, but not completely, counterbalanced by the higher levels of task identity and feedback from the job and the lower role ambiguity perceived by programmers. The general perception that analysis jobs are more prestigious than programming jobs could also contribute to higher work satisfaction among analysts.

Research Methods

In this section, the data collection and analysis procedures for testing the hypotheses presented in the previous section are described.

SUBJECTS AND PROCEDURES

Data were collected from 292 programmer/analysts at three locations (one in the Midwest and two in Canada) of a large manifacturing company. Thirty-eight subjects were excluded from the study: they were not programmer/analysts working for the participating company. This left a final sample of 254 programmer/analysts.

Participation in the study was voluntary. Subjects were contacted by the information systems management at the company. Approximately 80% of the programmer/analysts who were asked to participate in the study filled out the questionnaire. The questionnaire used to collect the data was administered in a classroom setting to groups of between ten and thirty subjects at a time and took about 45 minutes to complete.

MEASURES

For variables other than task differences and user interaction characteristics, standard measures found in the organizational behavior literature were employed. Hackman and Oldham's Job Diagnostic Survey (JDS) (Hackman and Oldham, 1980) provided measures of the five job characteristics. The role conflict and ambiguity scales were based on the eight and six item scales developed by Rizzo, House, and Lirtzman (1970) as modified by Bostrom (1981). The work satisfaction scale is taken from the Job Descriptive Index (JDI) developed by Smith, Cain, and Hulin (1969).

The user feedback scale consists of three items that are similar to the JDS's feedback from job scale. The user interaction scale consists of three items that are similar to the JDS's dealing with others scale. Goldstein (1985) contains a full description of both of the scales.

TASK DIFFERENCES MEASURES

Three scales were included to classify programmer/analysts as either developers or maintainers and as either

analysts or programmers: a rating scale, a 'last week' scale, and a 'last three months' scale. The contents of the scales can be found in Table 1.

In the rating scale, subjects were asked the extent to which their job involves each of a set of subtasks. The eleven subtasks used to measure maintenance/development differences were taken from the participating company's systems development procedures manual. Analysis/programming differences were measured with a scale used by Zmud and Baroudi (Baroudi, 1984).

The 'last week' and 'last three months' scales measured the amount of time spent by subjects on four maintenance/development tasks and three analysis/programming tasks in the previous week and in the previous three months of work. Tasks were combined from the subtasks used in the rating scales, because, in the pretest, subjects had difficulty allocating their time among a large number of subtasks.

Analysis of Task Differences Measures

The examination of the data indicated that there were two problems with our assumptions concerning the maintenance/development scales. First, the data did not indicate that production and customer support activities were subtasks performed by maintainers. In examining the ratings scale, there were strong correlations between the maintenance and enhancements items ($r=.78$) and between the customer and production support items ($r=.52$). The correlations between the maintenance and support items, however, were much lower (median $r=.33$). In addition, examination of the 'last week' and 'last three months' scales indicated that there were between 15 and 20 subjects classified as maintainers who spent all or almost all of their time on production or customer support activities.

There was also a problem with our assumptions concerning the analysis/programming scales. Two items did not clearly fit with the rest of either the analysis or programming items. In the rating scale, the program design and systems testing items correlated equally well with both the other analysis items and the other programming items. In addition, when examining the percentage scales, it became apparent that both analysts and programmers spent time in the systems testing activity.

Principal Measures of Task Difference

The results of the analysis discussed above were incorporated into the principal measures of task differences. The original idea of a maintenance/development measure was rejected and a maintenance/development/support measure was used in its place. In the ratings scale for this new measure the seven new systems development items were compared to the two maintenance items and to the two support items. For both percentage scales the new systems development percentage was compared to the maintenance and enhancements percentage and to the sum of the production and customer support percentages.

For the analysis/programming measure the program design and testing items were compared to four of the five analysis items (excluding system testing) to classify subjects based on the rating scale. For the percentage scales, the programming activities item was compared to the analysis activities item.

In classifying subjects using the ratings scales maximum ratings were compared instead of average ratings. It was found that the use of maximum ratings increased the agreement between the ratings classifications and the percentage classifications.

The three scales were combined into a single analysis/programming measure and a single maintenance/development/support measure. Table 2 presents data on the reliability of the scales. Reliability was calculated using the coefficient kappa (Kraemer, 1979). The results indicate that of the four scales, the combined scale is the most reliable and that of the three individual scales the "last three months' scale was the most reliable.

Second Set of Task Differences Measures

One problem not considered by the principal set of task differences measures is the separation of subjects who do some maintenance and some development from those who spend almost all of their time on either of these tasks. Task differences in work satisfaction and its determinants might be less pronounced among subjects who divide their time almost evenly over several tasks.

A second set of measures of task differences is included in this study so that differences among only subjects who had spent at least 80% of their time either programming or performing systems analysis and those who spent either 80% of their time maintaining systems, developing new systems, or supporting customers are considered. This second set of measures is called 'pure' measures. The first set of measures is referred to as 'overall' measures. The 'last three months' scale was used, because it was the most reliable of the three individual scales.

Task Differences Results

The last two sections set the stage for the analysis of the task differences results which are presented in this section. In the first subsection we will provide some prelim-

Table 1

Scales Used to Measure Task Differences

Rating Scales

To what extent does your job involve each type of task?

1	2	3	4	5
To a very little extent or not at all	To a little extent	To some extent	To a great extent	To a very great extent

1. New systems initiation
2. New system definition
3. New system proposal
4. New system specification
5. New system development
6. New system installation
7. New system evaluation
8. Maintenance of an existing system
9. Enhancements of an existing system
10. Production support
11. Customer support

1. Program design
2. Program coding
3. Program testing
4. Determine system requirements
5. System design
6. System testing
7. Evaluate an operational program/system
8. Evaluate system feasibilty

Last Week and Last Three Months Scales

Time spent in last week	Time spent in last three months	
_____%	_____%	1. New systems development
_____%	_____%	2. Maintenance/enhancements of existing systems
_____%	_____%	3. Production support
_____%	_____%	4. Customer support
100%	100%	
_____%	_____%	1. Programming activities (e.g., program design, program coding, program testing)
_____%	_____%	2. Analysis activities (e.g., determining systems requirements, system design, evaluating operational systems)
_____%	_____%	3. System testing
100%	100%	

Table 2a

Reliability of Scales Used in the Analysis/Programming Measures

	Rating Scale	Last Week Scale	Last Three Months Scale	Combined Scale
Rating Scale				
Last Week Scale	.61*			
Last 3 Month Scale	.67	.84		
Combined Scale	.70	.88	.96	

Table 2b

Reliabilty of Scales Used in the Development/Maintenance/Support Measures

	Rating Scale	Last Week Scale	Last Three Months Scale	Combined Scale
Rating Scale				
Last Week Scale	.51*			
Last 3 Month Scale	.56	.82		
Combined Scale	.63	.86	.93	

*—reliability calculated using coefficient kappa (Cohen, 1960)

inary information on our sample. In the second subsection differences between maintainers and developers, analysts and programmers and supporters and non-supporters (maintainers and developers) are presented. In the final subsection the relationship between the task dimensions is examined.

PRELIMINARY DATA ON THE SAMPLE

Three sets of preliminary information are provided in this subsection—demographic data for the entire sample, summary statistics on the dependent, task differences and explanatory variables, and demographic data for the different groups of programmer/analysts.

Several demographic variables provide a profile of the programmer/analysts in our sample. Of the 254 subjects, between 75% and 80% are under 40 years of age, with about equal numbers between 20 and 29, and between 30 and 39. Fifteen percent of the sample is between 40 and 49 and eight percent is 50 or over. With respect to level of education attained, 25% have a high school degree and possible some college education, 65% have a college degree and 10% have an advanced degree. The subjects' median time in the company is 5 years with a range of organizational experience being between 4 months and 40 years. The median information experience is 7 years with the range between 4 months and 37 years. The median time in the current job is 22 months with the range between 1 month and 19 years.

Table 3 presents summary statistics for our explanatory and dependent variables. The statistics for our explanatory variables indicate that the sample is similar to the four companies studied by Goldstein and Rockart (1984). All the internal consistency reliabilities for the explanatory and dependent variables are within the range deemed acceptable by Nunnaly (1978). In addition, the correlations between the explanatory variables and work satisfaction are all significant and all in the appropriate direction.

Table 4 presents the summary data fo rthe overall measure for both task dimensions. The crosstabulation indicates that a larger proportion of maintainers are programmers and a larger proportion of developers are analysts.

In our analysis of demographic differences in the task dimensions, we found that analysts are significantly older than programmers and have approximately two more years of information systems experience. Supporters are significantly older than non-supporters and have about 2.5 years more information systems experience. There are no significant differences in work satisfaction due to age or experience.

DIFFERENCES ALONG THE TWO TASK DIMENSIONS

Two statistical analyses are performed to study task differences. First, we examine differences in work satisfaction and in the explanatory variables using univariate analysis of variance. This allows us to test the hypotheses presented in the previous section.

Second, we examine overall differences in the means of the nine explanatory variables using profile analysis (Morrison, 1976). This technique consists of tests of hypotheses regarding three characteristics of the shape of the mean profiles for two groups of subjects. First, it tests whether the data support the hypothesis that the profiles of the means of the two groups are parallel. Since the hypotheses state that maintainers should rate themselves more highly on some explanatory variables than developers, we should expect that the mean profiles of the two groups would not be parallel. For similar reasons we would not expect the mean profiles for analysts and programmers to be parallel. If, however, we cannot reject the hypothesis that profiles are parallel, we can perform tests to determine if the profiles are either equal or flat.

Differences Between Developers and Maintainers

Table 5 presents the results of the analyses of variance involving both work satisfaction and the explanatory variables. The data provide strong support for our hypothesis (H9) that developers are more satisfied than maintainers.

Figure 2 presents the graphs of the mean profiles for developers and maintainers for the nine explanatory variables. The results of the profile analysis indicate that we must reject the hypothesis that the profiles are parallel ($F(8,189df)=2.49$, $p=.01$).

Insight into differences between maintainers and developers can be gained by examining the results for the individual explanatory variables. Using both the overall and pure measures, significant differences exist in skill variety in the direction predicted in H1 (skill variety is higher for developers). There is some support for our hypothesis (H2) that developers' jobs are more significant that those of maintainers. The results are in the appropriate direction, but are not statistically significant. For both autonomy and role conflict, differences exist but they are not in the direction predicted in the hypotheses.

Table 3

Summary Statistics for Dependent and Explanatory Variables

variable	number of items in measure	mean	std dev	internal consistency reliability	correlation with work satisfaction
skill variety	3	5.32	1.02	.66	.57
task identity	3	4.96	1.21	.70	.29
task significance	3	5.11	1.15	.72	.37
autonomy	3	5.12	.93	.71	.49
feedback from job	3	4.97	1.15	.76	.37
role conflict	6	3.20	1.08	.85	−.40
role ambiguity	8	3.68	1.18	.83	−.38
user feedback	3	4.32	1.16	.66	.38
user interaction	3	4.97	1.15	.75	.37
work satisfaction	18	5.12	1.02	.79	n/a

Notes: all variables are presented as if seven-point scales (satisfaction measure converted from three-point scales). reliability calculated using Cronbach's alpha.

Table 4

Categorization of Subjects Along Both Task Dimensions Using Overall Measure

	Developers	Maintainers	Supporters	Total
Analysts	19% (44)	6 (14)	6 (14)	31%
Programmers	14 (34)	46 (109)	9 (22)	69
Total	33%	52	15	100 (254)

Table 5

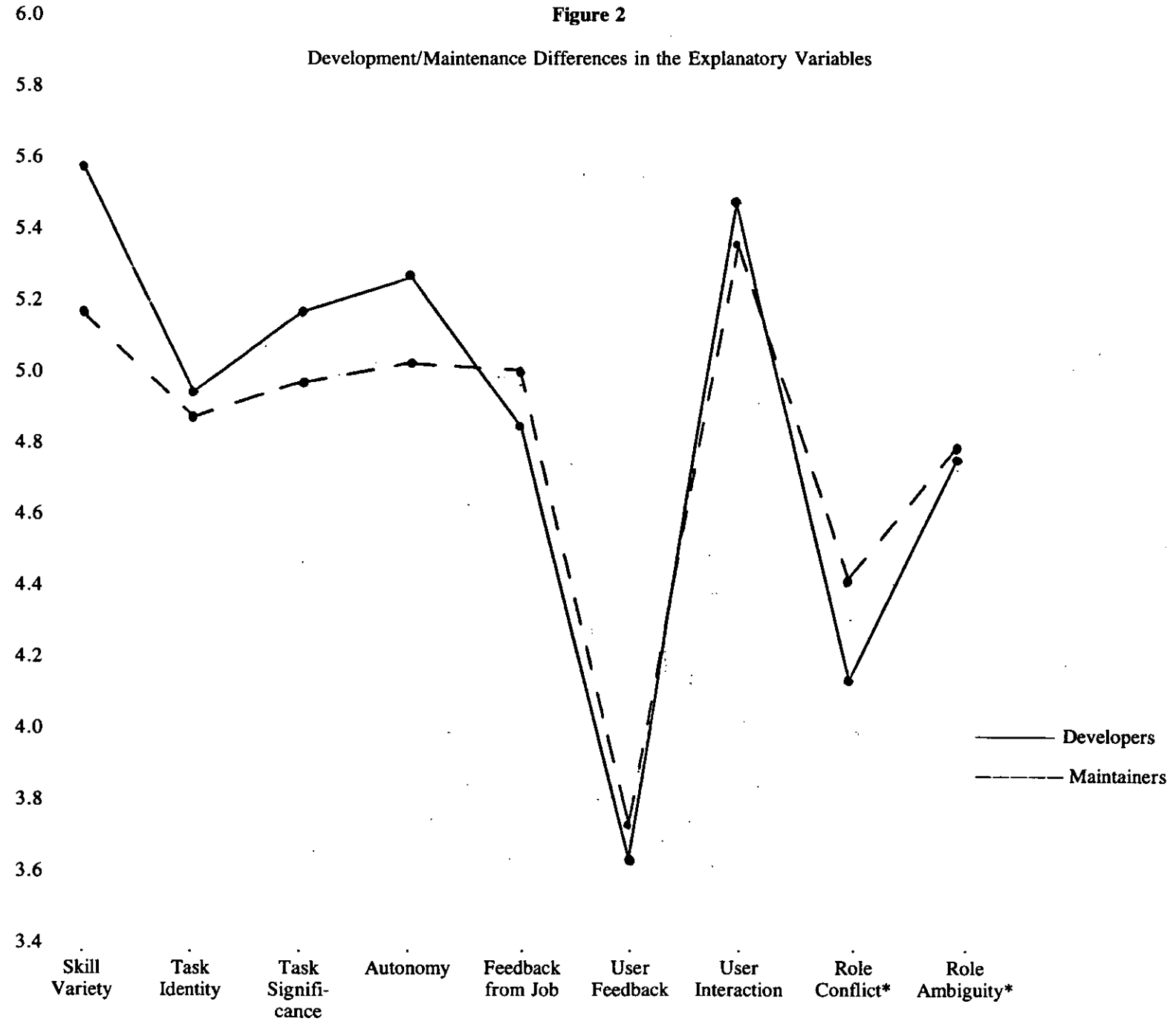
Differences Between Developers and Maintainers (Whole Sample)

	OVERALL MEASURE			PURE MEASURE		
	developers	maintainers		developers	maintainers	
variable	(n=78)	(n=128)	t-stat.	(n=49)	(n=63)	t-stat.
work satisfaction	5.31	4.99	2.27*	5.28	4.96	1.78*
skill variety	5.60	5.18	3.11*	5.41	5.02	2.23*
task identity	4.95	4.93	.33	4.80	4.96	.66
task significance	5.19	4.97	1.35	5.18	4.98	.92
autonomy	5.26	5.05	.83	5.20	4.89	1.80[1]
job feedback	4.88	5.01	.83	4.79	4.97	.80
user feedback	3.65	3.71	.26	3.45	3.48	.41
user interaction	5.50	5.45	.11	5.44	5.42	.10
role conflict	3.86	3.58	1.66[1]	3.82	3.45	1.52
role ambiguity	3.24	3.23	.07	3.28	3.22	.27

* — result significance at the .05 level in the predicted direction (one-tailed)
1 — the difference between groups is large, but it is not in the direction predicted by the hypothesis

Figure 2

Development/Maintenance Differences in the Explanatory Variables



*reverse score

That is, the data indicate that developers experience both more autonomy and more role conflict than maintainers (contradicting H3 and H8 respectively). There is no support for our hypotheses involving feedback from the job, user feedback, user interaction, or role ambiguity (H4, H5, H6, and H7, respectively). There are no differences between developers and maintainers in these variables.

Differences Between Analysts and Programmers

Table 6 presents the analyses of variance results for the analysis/programming differences. There are significant differences in work satisfaction, in the appropriate direction, when analysts and programmers are compared (confirming H19).

Figure 3 presents the graphs of the mean profiles for analysts and programmers for the nine explanatory variables. As in the first profile analysis, the results indicate that we must reject the hypothesis that the profiles are parallel ($F(8, 224df)=2.70$, $p=.01$).

In an examination of task differences in the explanatory variables, we can see that several of our hypotheses are confirmed. The data indicate that analysts interact more with users than programmers. These differences are significant when both the pure and overall measures are examined (confirming H16). In addition, there is evidence that analysts use more skills, view their task as more significant and are more autonomous than programmers, and that programmers receive more feedback from the job than analysts, partially confirming H10, H12, H13, and H14 respectively. These results are significant when one, but not when both, of the measures are examined. There is, however, no data to support the existence of differences between the two groups in task identity, user feedback, role conflict, or role ambiguity.

Differences Between Supporters and Non-Supporters

Table 7 presents differences between supporters and non-supporters for work satisfaction and the nine explanatory variables. The work satisfaction data indicate that there are only slight (non-significant) differences between supporters and non-supporters with the former group being more satisfied.

Figure 4 presents a graph of the profiles for supporters and non-supporters with respect to the nine explanatory variables. In contrast to the results of the previous two profile analyses, we cannot reject the hypothesis that the profiles are parallel ($F(8,231df)=1.30$, $p=.24$). We must, however, reject both the hypotheses that the profiles are at a similar level ($t(238df)=2.04$, $p=.04$) and that they are flat ($F(8, 237df)=59.9$, $p=.00$). These results are explained by an examination of Figure 4. The figure indicates that the differences between supporters and non-supporters are large and relatively uniform and that there is a large variation in the combined means of the two groups.

When considering the measures individually, there are significant differences between supporters and non-supporters on only three of the explanatory variables—autonomy, user feedback and user interaction.

COMPARISON OF FIVE GROUPS OF PROGRAMMER/ANALYSTS

In the preceding subsections, we examined differences along the two original task dimensions and differences between supporters and non-supporters. Our sample is split into five groups—development/programmers, development/analysts, maintenance/programmers, maintenance/analysts and supporters in this subsection. This categorization allows us to make a more detailed examination of programmer/analysts. For example, we can examine whether analysts are more satisfied with their work than programmers when only developers or only maintainers are considered. In addition, we can compare supporters to each of the other four groups of programmer/analysts, not just to non-supporters as was done in the previous section.

Two statistical analyses will be used to study the five groups. First, analysis of variance involving all five groups of subjects will be used to determine if overall differences among the groups exist for work satisfaction and for our explanatory variables. Second, for each variable a series of pairwise analyses of variance will be carried out to determine if significant differences exist between each of the pairs of groups of programmer/analysts.

Table 8 presents the mean scores for the five groups. With respect to work satisfaction, the differences among the five groups approach significance ($p=.09$). An examination of the scores indicates that maintenance/programmers are less satisfied with their work than any of the other four groups. The data also indicate that the other four groups are approximately equally satisfied with their work. The results of the pairwise comparisons indicate that maintenance/programmers are significantly less satisfied than both development/programmers and development/analysts.

With respect to the job characteristics there were significant differences among the five groups in two of the

Table 6

Differences Between Analysts and Programmers (Whole Sample)

	OVERALL MEASURE			PURE MEASURE		
variable	analysts (n=74)	programmers (n=167)	t-stat.	analysts (n=38)	programmers (n=75)	t-stat
work satisfaction	5.30	5.06	1.74*	5.51	4.97	2.86*
skill variety	5.46	5.32	1.08	5.55	5.24	1.67*
task identity	4.89	5.01	.69	5.03	4.95	.33
task significance	5.20	5.07	.80	5.35	4.96	1.70*
autonomy	5.32	5.12	1.54	5.40	5.00	2.17*
job feedback	4.73	5.12	2.57*	4.85	5.00	.61
user feedback	3.81	3.80	.03	4.00	3.68	1.04
user interaction	5.73	5.43	2.26*	5.85	5.45	2.24*
role conflict	3.87	3.60	1.64	3.76	3.39	1.56
role ambiguity	3.33	3.16	1.13	3.20	3.07	.62

Table 7

Comparison of Support Personnel to Non-Support Personnel

	OVERALL MEASURE			PURE MEASURE		
variable	support (n=42)	non-support (n=206)	t-stat.	support (n=14)	non-support (n=151)	t-stat
work satisfaction	5.26	5.15	.66	5.51	5.20	1.25
skill variety	5.31	5.18	.50	5.52	5.35	.68
task identity	5.06	4.95	.54	4.86	5.00	.39
task significance	5.42	5.08	1.78	5.55	5.13	1.43
autonomy	5.47	5.15	2.03*	5.67	5.11	2.14*
job feedback	5.10	4.94	.85	4.98	4.97	.01
user feedback	4.32	3.68	2.50*	5.07	3.63	3.31*
user interaction	5.76	5.47	1.77*	5.95	5.43	1.97
role conflict	3.64	3.72	.40	3.70	3.56	.41
role ambiguity	3.04	3.23	1.05	2.87	3.20	1.04

* — result significant at the .05 level (two-tailed)

Table 8

Differences in the Dependent and Explanatory Variables
Among the Five Groups of Programmer/Analysts

variable	supporters (n=42)	development analysts (n=44)	development programmers (n=34)	maintenance analysts (n=16)	maintenance programmers (n=109)	F-statistic (4,240df)
work satisfaction	5.26	5.29[a]	5.34[b]	5.32	4.94[ab]	2.05
skill variety	5.31	5.52	5.71[a]	5.35	5.17[a]	2.46*
task identity	5.06	4.95	4.95	4.73	4.98	.21
task significance	5.42[a]	5.15	5.25	4.96	4.96[a]	1.44
autonomy	5.47[a]	5.26	5.25	5.35	4.99[a]	2.45*
job feedback	5.10	4.64[a]	5.19[a]	5.10	5.02	1.57
user feedback	4.32[ab]	3.73	3.54[a]	3.62	3.74[b]	1.61
user interaction	5.76[abc]	5.83[def]	5.09[ad]	5.21[be]	5.48[cf]	4.23*
role conflict	3.64	3.98[a]	3.71	3.69	3.55[a]	1.05
role ambiguity	3.04[a]	3.58[abc]	2.80[bd]	2.91[c]	3.28[d]	3.12*

* — significant at the .05 level (two-tailed)
abcdef — significant differences between pair of variables (two-tailed at the .05 level)

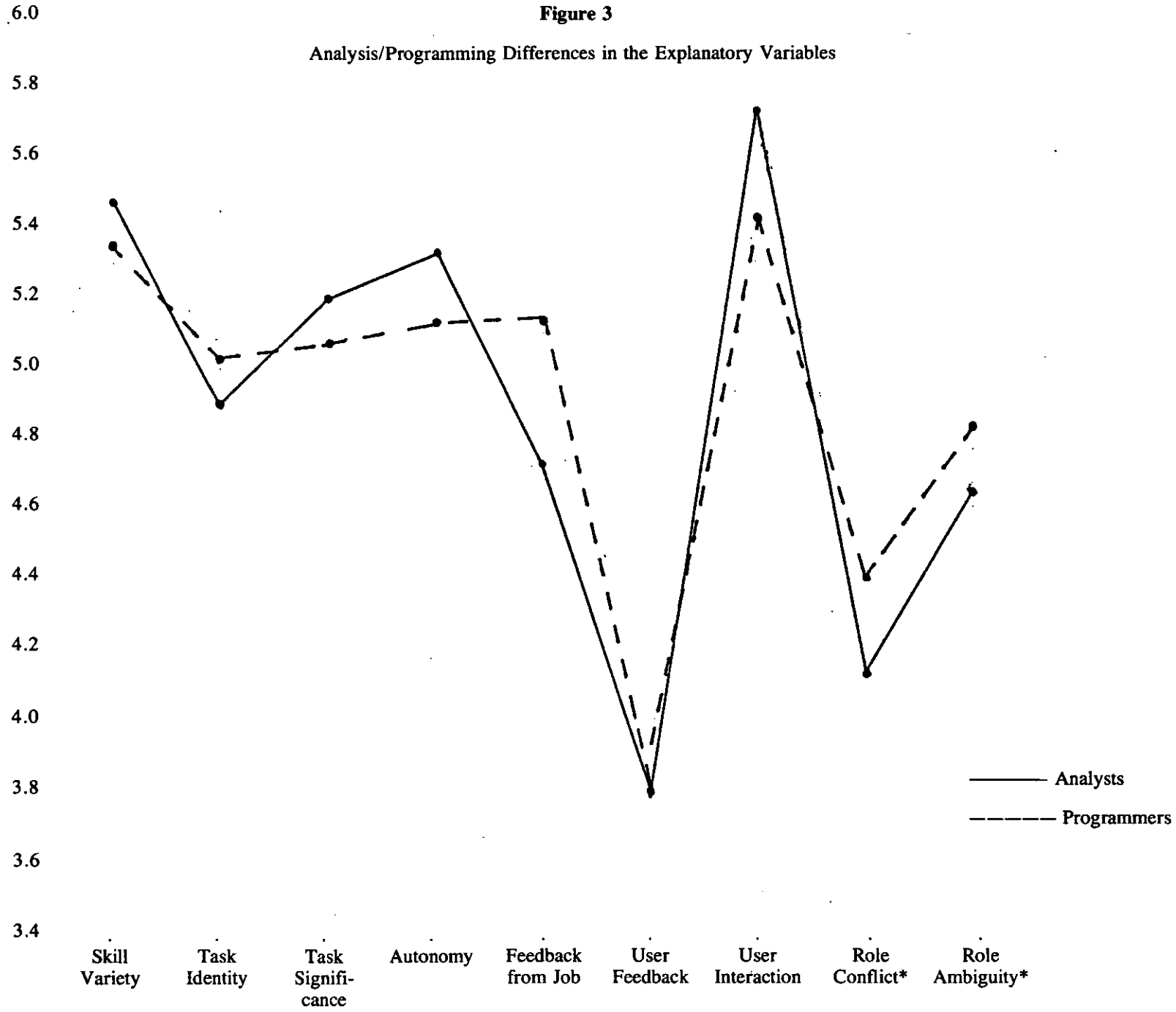
Table 9a

Summary of Tests of Differences Between Maintainers and Developers
for the Dependent and Explanatory Variables

Hypothesis		Better for Developers	About the Same	Better for Maintainers
1	Skill variety	$c^{1,2}$		
	Task identity		$c^{1,2}$	
2	Task significance	n^3		4
3	Autonomy			n^4
4	Feedback from job			n^3
5	User feedback			n^3
6	User interaction			n^4
7	Role ambiguity	n^4		
8	Role conflict	n^4		
9	Work Satisfaction	$c^{1,2}$		

Figure 3

Analysis/Programming Differences in the Explanatory Variables



*reverse score

Figure 4

Support/Non-Support Differences in the Explanatory Variables

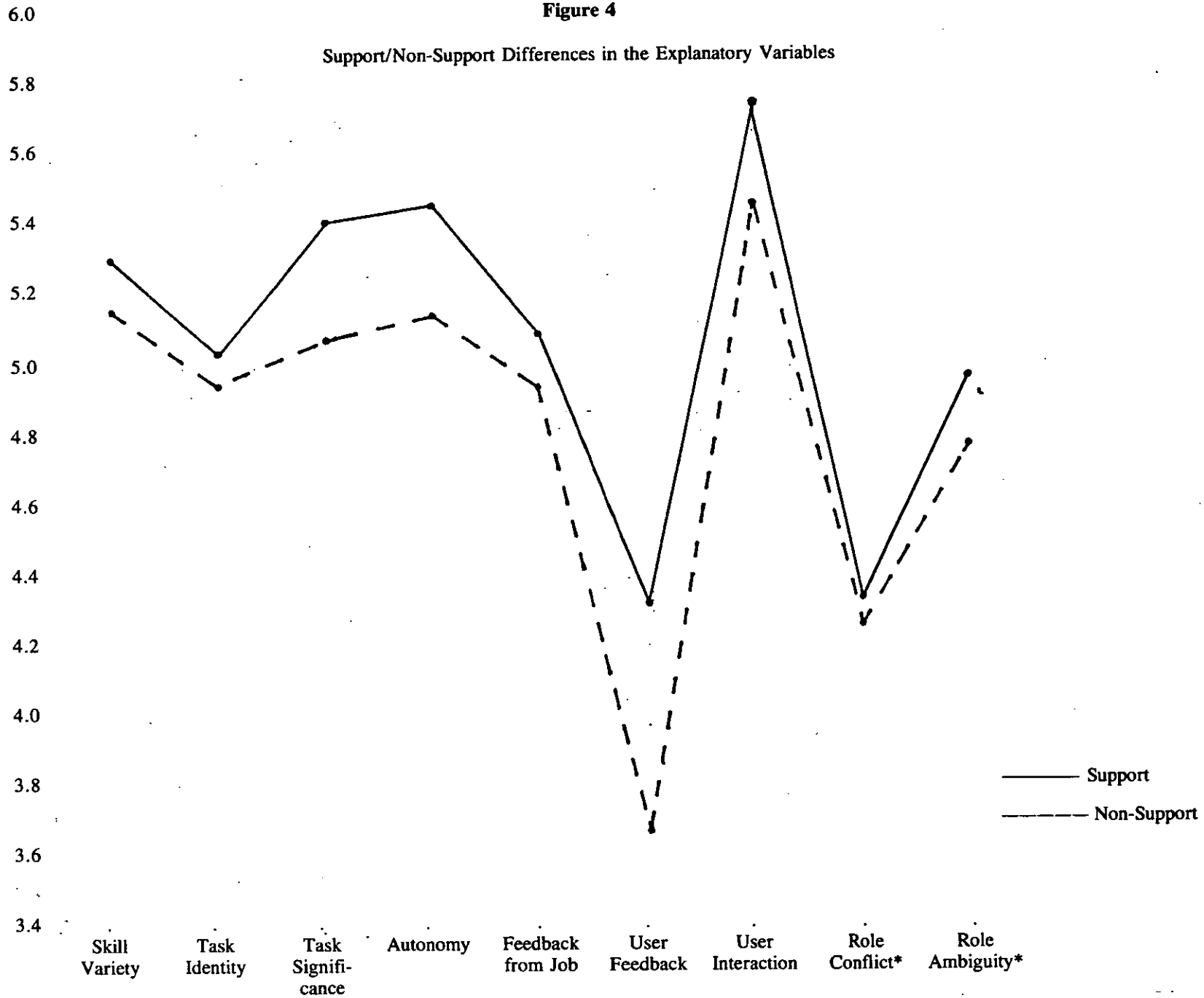


Table 9b

Summary of Tests of Differences Between Analysts and Programmers for the Dependent and Explanatory Variables

Hypothesis		Better for Analysts	About the Same	Better for Programmers
10	Skill variety	c^2		
11	Task identity			n^5
12	Task significance	c^2		
13	Autonomy	c^2		
14	Feedback from job			c^1
15	User feedback	n^3		
16	User interaction	$c^{1,2}$		
17	Role ambiguity			n^3
18	Role conflict	n^4		
19	Wokrk Satisfaction	$c^{1,2}$		

key:

$c^{1[2]}$ — hypothesis confirmed when overall (pure) measure is used
$c^{1,2}$ — hypothesis confirmed when either measure is used
n^3 — hypothesis not confirmed, but results are in the appropriate direction when either measure is used
n^4 — hypothesis not confirmed and results are in the direction opposite to the hypothesized direction when either measure is used
n^5 — hypothesis not confirmed and results are in opposite directions when the different measures are used

variables—skill variety and autonomy. The pairwise comparisons indicated that maintenance/programmers used significantly fewer skills than both groups of developers and that they were less autonomous than supporters.

In examining the user interaction characteristics and role perceptions there were significant differences among the five groups in two of the four variables—user interaction and role ambiguity. Both supporters and development/analysts interacted more with users than both development/programmers and maintenance/analysts. Further, supporters interacted more with users than maintenance/programmers, who interacted more with users than development/programmers. Development/analysts experienced significantly more role ambiguity than all of the other groups except maintenance/programmers and maintenance/programmers experienced more role ambiguity than development/programmers.

In addition, there were three significant pairwise ANOVAs involving other variables. Development/programmers experienced significantly more feedback from their jobs than development/analysts. Supporters received more feedback from users than development/programmers. Development/analysts experienced more role conflict than maintenance/programmers. The results for these other variables should be interpreted very cautiously. By chance, we would expect several significant results to occur when we carry out a large number of pairwise comparisons.

Discussion

The data presented above support the hypotheses that task differences in work satisfaction exist when maintainers and developers, and when analysts and programmers are

compared. In addition, the data support some of the hypothesized differences in the explanatory variables. In this section the research results are summarized, a revised task differences model is presented, and the implications of the research for managers and researchers are outlined.

Table 9 summarizes the results of the tests of our hypotheses. The comparison of maintainers and developers presented in Table 9a indicates that developers are significantly more satisfied with their work than maintainers, confirming H19 and the results obtained by Baroudi and Ginzberg (1984). When comparing these two groups with respect to the nine explanatory variables only one of these hypotheses (H1: skill variety is higher for developers) is supported by the data. Results are in the appropriate direction when three other variables are considered—task significance (higher for developers), feedback from the job and user feedback (both higher for maintainers). For two other variables, however, there is evidence that significant differences exist in the opposite direction to that stated in the hypotheses. The data indicate that developers are more autonomous and experience more role conflict than maintainers.

A larger number of hypotheses are confirmed when analysis/programming differences are examined (Table 9b). As predicted in H19 and as observed by Couger and Zawacki (1980), analysts are more satisfied with their work than programmers. There are significant differences in the hypothesized direction when analysts and programmers are compared with respect to five of the nine explanatory variables—skill variety, task significance, autonomy, user interaction (all higher for analysts) and feedback from the job (higher for programmers). Differences between the two groups are in the appropriate direction, but are not significant, for two other variables (user feedback and role ambiguity). There is little difference between the groups in task significance. When role conflict is examined the data provide evidence that programmers experience more conflict than maintainers, contradicting H18 and the results presented by Baroudi (1984).

In addition to the tests of the hypotheses, several other observations can be made based on the results. First, the data provide evidence of interaction effects between the two task dimensions when work satisfaction is examined. The significant difference in satisfaction between maintainers and developers and between analysts and programmers can be attributed to the low score on work satisfaction among maintenance/programmers. Maintenance/analysts, development/analysts, and development/programmers have almost identical work satisfaction scores.

Second, there is evidence of interaction effects when some of the explanatory variables are examined. For example, there is almost no difference in role ambiguity when all developers and all maintainers are compared. When programmers alone are considered, however, developers experience significantly less ambiguity than maintainers; the reverse is true when analysts alone are considered (see Table 8).

Third, a group of programmer/analysts not considered in our original model, supporters, exists at the company we studied. This group is different from other programmer/analysts in the type of work they do and in the characteristics both of their work and of their interactions with others.

Fourth, the model described in this article might contain too many variables. There is evidence that task differences exist in only a subset of the variables included in our study. It might be feasible, therefore, to use a more parsimonious model of task differences.

Given the findings discussed above, several changes are needed to the model of task differences proposed in this article. First, the two task dimensions should be replaced with five groups of programmer/analysts. This change is warranted because of the interaction effects between the two task dimensions and because of the presence of the support group. Second, fewer explanatory variables should be included in the model. When the five groups of programmer/analysts are examined differences exist in only four of the explanatory variables.

The revised model can be represented as a set of rankings of the five groups with respect to work satisfaction and the four explanatory variables (see Table 10). The rankings are based on the differences found in the data. The model hypothesizes that maintenance/programmers are the least satisfied group of programmer/analysts and that the other four groups are approximately equally satisfied. With respect to the explanatory variables it is hypothesized that maintenance/programmers will have low levels of skill variety and autonomy and moderate levels of role ambiguity and user interaction. Supporters, in contrast, will have high levels of autonomy and user interaction, low levels of role ambiguity and a moderate level of skill variety. The other groups have a more mixed set of scores for the explanatory variables.

The task differences in the explanatory variables could be viewed as causes of the task differences in work satisfaction. The relatively low scores for the maintenance/programmers on the explanatory variables could be the cause of their poor job satisfaction. The relatively higher scores for the development/analysts, development/programmers and maintenance/analysts could cause their higher satisfaction.

The explanatory variables, however, provide no insight into the satisfaction score of the supporters. Given that

Table 10

Hypothesized Ranking of the Five Groups of Programmer/Analysts with Respect to Certain Variables

variable	supporters	development analysts	development programmers	maintenance analysts	maintenance programmers
work satisfaction	H	H	H	H	L[1]
skill variety	M	H	H	M	L
autonomy	H	M	M	M	L
user interaction	H	H	L	L	M
role ambiguity[2]	H	L	H	H	M

key:

H — high
M — moderate
L — low

[1] — maintenance/analysts should be significantly less satisfied then any of the other groups; there should be little difference in satisfaction among the other groups
[2] — reverse scored

this group scores relatively highly on the explanatory variables we would expect them to be more satisfied than both the two groups of developers and the maintenance/analysts. There might be some explanatory variables, not included in this study, that could lead to a reduction in the work satisfaction of supporters.

The research results provide guidance for managers interested in redesigning the work of programmer/analysts so that work satisfaction is improved by providing insights into the relative satisfaction of different groups of programmer/analysts and into the relative strengths and weaknesses of the characteristics of their work. If managers at the company studied were to apply the results they would focus their redesign efforts on improving the work of maintenance/programmers. These workers are both the largest and least satisfied groups of subjects. The redesign of maintenance/programmers' jobs would concentrate on improving their autonomy and on increasing the number of skills they use in their work. These are areas of relative weakness for maintenance/programmers. This recommendation must be applied cautiously because the data on which these recommendations were made were collected at only one company.

The results of this study point up the need for a more detailed study of the support role. Several researchers have noted that increases in both end user computing and in the use of personal computers will lead to an increase in the need for supporters (Rockart and Flannery, 1983; Quillard, Rockart, Vernon, Mock, and Wilde, 1983). One extension to this study would involve interviewing supporters to determine the strengths and weaknesses of their work. A study could then be carried out and recommendations could be made for improving the work satisfaction of these individuals.

In addition, the results point up the need for qualitative research that addresses the relationship between organizational strategies and work satisfaction in programmer/analysts. Companies take many different approaches to organizing their information systems function. For example, some have separate development and maintenance staffs while others organize their staffs with all the developers and maintainers for a given user group working together. Case studies of companies that are successful in managing their information systems development and maintenance staffs could provide important insights into these organizational issues.

The findings of this research have significant implications for both managers and researchers. Principally, the observations that programmer/analysts cannot be treated as a homogeneous group and that differences in tasks are important determinants of work satisfaction affect the ways in which managers should consider redesigning programmer/analysts' work. In addition, the task differences in work satisfaction and in several of the explanatory variables between supporters and more traditional information systems workers provide important insights for those interested in the effects of the changes that will occur in the information systems function.

REFERENCES

Baroudi, J.J., "The Impact of Role Variables on Information Systems Personnel Outcomes," Center for Research on Information Systems Working Paper 81, New York University, New York, New York.

Baroudi, J.J., Ginzberg, M.J. "Impact of the Technological Environment on Programmer and Analyst Job Outcomes," Center for Research on Information Systems Working Paper 85, New York University, New York, New York.

Bostrom, R.P. "Role Conflict and Ambiguity: Critical Variables in the User-Designer Relationship," *Proceedings of the Seventeenth Annual Computer Research Conference*, 1981, 88-112.

Cohen, J. "A Coefficient of Agreement for Nominal Scales," *Educational and Psychological Measurement*, Volume 20, 1960, 37-46.

Couger, J.D., Cotler, M.A., "The Effects of Maintenance Assignments on Goal Congruence for Programmers and Analysts," *Proceeding Conference on Information Systems*, 1983, 83-100.

Couger, J.D. and Zawacki, R. *Motivating and Managing Computer Personnel*, John Wiley and Sons, New York, New York, 1981.

Fjelstad, R.K., Hamlen, W.T. "Application Program Maintenance Study-Report to Our Respondents," *Proceedings GUIDE 48*, Philadelphia, Pennsylvania, 1979.

Goldstein, D.K., "Work-Related Determinants of Job Satisfaction in Programmer/Analysts: A Comprehensive Model and an Examination of Task Differences," Unpublished Doctoral Dissertation, Massachusetts Institute of Technology, 1985.

Goldstein, D.K. and Rockart, J.F. "An Examination of Work-Related Correlates of Job Satisfaction in Programmer/Analysts," *Management Information Systems Quarterly*, Volume 8, 1984, 103-115.

Hackman, J.R. and Oldham, G.R. "Motivation Through the Design of Work: A Test of a Theory," *Organizational Behavior and Human Performance*, Volume 16, Number 2, August 1976, pp. 250-279.

Hackman, J.R. and Oldham, G.R. *Work Redesign*, Addison Wesley, Reading, Massachusetts, 1980.

Kraemer, H.C. "Ramifications of a Population Model for K as a Function of Reliability," *Psychometrica*, Volume 44, 1979, 461-472.

Morrison, D.F. *Multivariate Statistical Methods*, Second Edition, McGraw-Hill, New York, 1978.

Nunnally, J.C. *Psychometric Theory*, Second Edition, McGraw-Hill, New York, 1978.

Quillard, J.A., Rockart, J.F., Wilde, E., Vernon, M., Mock, G., "A Study of the Corporate Use of Personal Computers," Center for Information Systems Research Working Paper 109, Cambridge, Massachusetts, 1983.

Rizzo, J.R., House, R.J., and Lirtzman, S.J. "Role Conflict and Ambiguity in Complex Organizations," *Administrative Science Quarterly*, Volume 15, Number 2, June 1970, pp. 150-163.

Rockart, J.F., Flannery, L.S. "The Management of End User Computing," *Communications of the ACM*, Volume 26, Number 10, 1983, 776-784.

Walston, C.E., Felix, C.P. "A Method of Programming Measurement and Estimation," *IBM Systems Journal*, Volume 16, 1977, Number 1, 54-73.