Using Belbin's Roles to Improve Team Effectiveness:

An Empirical Investigation

K. Todd Stevens* **and Sallie M. Henry**

Department of Computer Science
Virginia Tech
Blacksburg, VA 24061

Abstract

This paper presents a controlled experiment conducted with software engineering students that demonstrates the utility of forming teams based on R. Meredith Belbin's set of team roles. The overall research effort is a demonstration of the general utility of Belbin's roles in improving the effectiveness of teams, even industry teams. The significance of this work is twofold: performance and team viability. Performance improvements clearly improve a team's productivity; viability issues are important because if employees remain with a team or employer, then employee replacement costs are reduced. To address this problem, as an initial step, controlled experiments have been conducted to demonstrate that teams that contain certain roles perform better than teams that do not. In a laboratory setting, a number of teams were formed that contained a single leader; other teams were formed that had no leader or multiple leader. The results of this single experiment are positive. They demonstrate that indeed Belbin's roles are useful knowledge in forming teams. The specific conclusion of this first controlled experiment is that a single leader on a team makes a team perform better than having multiple leaders or no leader. In other words, as one would expect, the mean time to completion for the leaderless group of teams was significantly larger than the group of teams with leaders. This means that Belbin's roles can be utilized in team formation, making sure that a team has a single leader, and also for evaluation of extant teams. Both of these aspects, formation and evaluation, are extremely useful to managers of software programmers.

1. Introduction

The Software Engineering community is always in crisis. As Brooks [Broo87] says, there is no "silver bullet" to resolve the monumental problems that software engineers must address, so that the best that one can realistically hope to achieve is improvement. There is always a great deal of ongoing research to try to reduce the amount of code that developers need to produce for a product, e.g. through reusability, or to improve the code or the amount of code that individuals produce. This research focuses on improving the productivity of teams.

Some research, for example the Capability Maturity Model (CMM) by the Software Engineering Institute (SEI) [Hump89, Hump95, Paul95], has focused on defining levels of **capability** that organizations can achieve for their software development **process**. Unfortunately, just because a team's capability is better than another team's does not mean that the team is actually performing better. In the authors' opinion, an organization or specific team that is better than another should demonstrate one or both of the following: process improvement or product improvement. One issue that this research addresses is the direct measurement of the performance of software teams, not their capability and not their organization's effectiveness.

A second, more concrete issue motivating this research is cost. The cost of developing software is always on the rise, and cost control and even reduction have

always been of great concern. One part of the cost to develop software is what DeMarco calls “manpower expense” [DeMa87], which includes the costs of old employees passing their knowledge on to other employees, training new employees, etc. Even in 1981, a modest estimate of the overhead costs to replace a typical programmer was $10, 000 [Cher81]; DeMarco estimates that “turnover costs about twenty percent of all manpower expense” [DeMa87]. How can these non-productive additional costs of software development be reduced or controlled? One way is to make sure that employees are happy at their work and therefore elect to remain with one employer [DeMa87]. Even in 1981, Boehm felt that personnel issues would eventually be acknowledged as a key issue in productivity [Boeh81], not language and implementation issues that are currently prevalent in research. This research presents a novel approach to addressing this issue: assign personnel to teams based on who can work best together. Employees who work well together will be happier, more productive, and stay with the company longer, thereby reducing the non-productive turnover costs by ridding the workplace of interpersonal conflict.

This research investigates how to improve the effectiveness of software development teams in an innovative manner: form teams based on who can work effectively together. The **effectiveness** of a team is described in the management literature in terms of two aspects: **performance and viability** [Sund90]. Performance consists of customer satisfaction, schedule and cost adherence, etc. Viability “entails members’ satisfaction, participation, and willingness to continue working together” [Sund90].

These definitions correspond to the two problems described above: capability issues and manpower costs.

2. Background and Motivation

Interdisciplinary research has been investigated for decades. In the 1970s, Weinberg discussed an interdisciplinary approach to software development and psychology [Wein71]. In the 1980s, the realization that software could not keep pace with hardware "focus(ed) attention on human factors in the process of system development as well as the performance of the end user of computer systems" [Nich82].

This investigation takes an interdisciplinary approach to address the question of how to assess the effectiveness of software development teams by examining teams from the prospective of various disciplines, much the same way as Curtis has encouraged [Curt86]. He mentions several factors that might be relevant to such interdisciplinary research including psychological factors, the transfer of skills from other domains, and communication issues. The approach taken for this investigation includes perspectives from psychology and management to computer science personnel applying empirical studies.

2.1 Empirical Studies

Software engineering has been using measurement since the 1970s [Cont86] and has spawned powerful support and interest in both academia and industry to such an

extent that many software development organizations, particularly ones that deal with the federal government, require a measurement or metrics program.

Initial interest in the current investigation began with an interest in quantitative research and empirical studies, which lead to the discovery of the Empirical Studies of Programmers that occurs about every two years [Solo86, Olso87, Koen91, Cook93]. The proceedings for the workshops contain very diverse and interesting descriptions of studies, with qualitative and quantitative conclusions, that are derived from and useful to software developers.

These empirical studies take on a variety of forms, based primarily on the type of data being evaluated and the type of results anticipated, e.g. conceptual models, quantitative models, or identifying trends. Some empirical research focuses on the actions and verbal queues expressed by developers [Walz87, Holt87, Curt90, Flor91, Stra94], sometimes defining categories for actions or verbal queues. Work such as these tends to be interested in formulating a model to explain observed behaviors. Other research focuses on "hard" objective, numeric data that might be collected automatically, for example through modifications to compilers or editors to count number of modifications, amount of time to accomplish a modification, etc. [Henr93]. This type of work tends to produce results with general applicability that is difficult to dispute because of its objective foundation. This previous work, done by one of the authors, uses a combination of objective data in tandem with subjective questionnaire data; it demonstrates empirically that an object-oriented language is easier to use and makes individuals more productive.

In another empirical investigation, Henry and Lewis gathered data on programmers who reuse software components, how language paradigms affect reuse and how encouraging programmers to reuse affects reuse [Lewi92].

The most interesting of the empirical studies focuses on more personal information such as personality attributes [Bost81, Nels91], which lead this research towards an investigation of personality characteristics, which in turn lead to roles. Consequently, this research investigates the empirical relationships between roles of team members and team effectiveness.

2.2 Psychology Background

This investigation began with an interest in personality characteristics, leading to the Myers-Briggs Type Indicator (MBTI) [Myer80, Myer85, Myer87], which ultimately plays only a minor role in this investigation. As the research progressed, we determined that a slightly different approach might be more effective, something more directly applicable to investigating teams. What we discovered was Belbin's roles in the literature on team evaluation [Belb81, Belb93]. Therefore, the basic concepts of roles in the arena of psychology are described, after which we present an outline of Belbin's specific roles that are used in this study.

Role theory, functions of roles, and role complements are important parts of this research. *Role theory* is defined in different ways by various authors; for example Biddle, from which much of this discussion is drawn, describes it as "behaviors characteristic of one or more persons in a context" [Bidd79]. Sarbin takes a particularly interesting view

of role theory because he specifically includes personality in his description: role theory "is an interdisciplinary theory in that its variables are drawn from studies of culture, society, and personality" [Sarb54]. Role theory is founded in the meeting of two different theoretical trends in sociology: symbolic interactionism and structural-functionalism. The former is unimportant to this research, but structural-functionalism concerns the idea that "actions are patterned into coherent and ordered systems that govern both interpersonal interaction and societal functioning" [Know82], which introduces the idea of functions.

The characteristic behaviors mentioned above can be defined as *functions* associated with a given role. Biddle states that role functions are a "behavioral repertoire characteristic of a person or position" [Bidd79], in other words a role. He further states that roles can be viewed in terms of their consequences, which are their "characteristic effects, or functions, within a societal system" [Bidd79].

Two other concepts in the role theory field are *functional requisite* and *role complement*, which are essentially the foundation of Belbin's work [Belb81] that is an essential component of this research and is presented in Section 2.3. Functional requisites are "functions that are necessary for the survival or maintenance of a social system" [Bidd79], and the survival of a team can be considered its success or effectiveness. Role complement is a set of roles defined for a given context. These two concepts combine to form the functional requisites that are embodied by Belbin's roles, which is an example of a team complement.

2.3 Management Background

The significance of this work lies in the applicability to the individuals who form teams: managers. Hence, the ultimate focus of this research is on the importance to managers in forming or evaluating teams. The following sections present the Belbin roles, from the management literature, that are used in this empirical study.

Belbin's Roles

R. Meredith Belbin conducted a series of experiments that produced results that are the very foundation of this research. His results consist of a model of **management teams** based on roles that need to be present for a management team to be successful, in other words a role complement. He started with a simple idea that different types of people interact in different ways. Initially, he investigated teams composed of members who were all very similar. Then, he conceived of various other types through extensive observation of teams working together. For example, one such type was labeled a "Plant" because this type of team member appeared to "sit in the corner" and not interact a lot, like a house plant, but presented very innovative plans and ideas when she or he did interact. Eventually, through observation, hypothesis testing, and experimentation, Belbin defined eight roles based on these observed "types" that he felt were necessary for a team to be successful: Chairman, Shaper, Plant, Monitor-Evaluator, Resource Investigator, Team Worker, Company Worker, and Completer-Finisher. His work is described in [Belb81, Belb93].

Table 2.3.1 provides a brief overview of all of the Belbin roles. It is copied from Belbin's original presentation of the material [Belb81]. Appendix A contains more complete descriptions of the roles.

Table 2.3.1: Descriptions of Belbin Roles

Name	*Symbol*	*Typical Features*	*Positive Qualities*	*Allowable Weaknesses*
Company Worker	CW	Conservative, dutiful, predictable	Organizing ability, practical common sense, hard-working, self-discipline	Lack of flexibility, un-responsiveness to unproven ideas
Chairman	CH	Calm, self-confident, controlled	A capacity for treating and welcoming all potential contributors on their merits and without prejudice. A strong sense of objectiveness	No more than ordinary in terms of intellect or creative ability
Shaper	SH	Highly strung	Drive and a readiness to challenge inertia, ineffectiveness, complacency or self-deception	Proneness to provocation, irritation and impatience
Plant	PL	Individualistic, serious-minded, unorthodox	Genius, imagination, intellect, knowledge	Up in the clouds, inclined to disregard practical details or protocol
Resource Investigator	RI	Extroverted, enthusiastic, curious, communicative	A capacity for contacting people and exploring anything new. An ability to respond to challenge	Liable to lose interest once the initial fascination has passed
Monitor-Evaluator	ME	Sober, unemotional, prudent	Judgement, discretion, hard-headedness	Lacks inspiration or the ability to motivate others
Team Worker	TW	Socially oriented, rather mild, sensitive	An ability to respond to people and to situations, and to promote team spirit	Indecisiveness at moments of crisis
Completer-Finisher	CF	Painstaking, orderly, conscientious, anxious	A capacity for follow-through, perfectionism	A tendency to worry about small things, a reluctance to "let go"

While these roles may be interesting, the question arises "How can these roles be measured for an individual?" Belbin developed a questionnaire as part of his investigations, which can be found in [Belb81]. This is the instrument used to gather data from individuals to analyze the teams.

2.4 Using Student Programming Teams

The subjects in the controlled experiments that form the core of this paper are senior college students. Other investigations have looked at student programming teams. Henry evaluates various methods used to set up programming teams for collegiate senior-level software engineering classes [Henr83]: random, letting them set up own, grade point, etc. She presents a heuristic for establishing teams that is based on amount of free time, schedule conflicts, and grade point that that achieves relatively equivalent teams. Scott and Cross also discuss issues in setting up student programming teams, in an effort to make them relatively equivalent [Scot95]. Some of the issues that they present are academic performance (grades), team and project size, and, interestingly enough, psychological profiles including the Myers-Briggs Type Indicator (MBTI), a popular personality psychometric test. Unfortunately, their treatment is very superficial; they only consider issues such as a team needing an introvert and an extrovert because the class requires written and oral presentations, and *clearly*, they conclude, "an Extrovert will be more comfortable with an oral presentation, while an Introvert may produce a better written report" [Scot95].

Finally, the issue of using student programmers in experiments introduces the concern that such research does not directly apply to industry programmers. Holt, et al. demonstrate that advanced students and professional programmers are statistically similar in terms of comparing their mental representation and various performance measures

[Holt87]. This provides support for using students in studies, especially for initial investigations where an industry validation is to be done.

2.5 Management Team Research Applied to Software Teams

As presented in previous sections, many studies have been done on software development teams, e.g. [Jeff87, Mill83, Bost81, Shne80, Trow91, vonM84, Walz87, Zahn90]. Some of this work has management overtones, others not. One study by Rob Thomsett is of particular interest for this investigation because it applies Belbin's roles to software developers. His study presents a qualitative analysis of software teams in Australia using three measurement instruments for the three different models [Thom90]: the MBTI, the Belbin Self-Perception Inventory, and the Job Diagnostic Survey (JDS).

For one company that participated in Thomsett's workshop, "Immediate productivity increases of 200 percent (were) reported by the senior management of the computing group" [Thom90]. Thomsett describes some intuitive reasons that the Belbin roles, as well as the MBTI types, are distributed the way they are in his data: a 'cloning' effect that means that managers hire employees that are similar to themselves. Further, he observes that "At best because of the relative lack of Team Worker … and effective Chairpersons …, computing teams and managers generally lack the required interpersonal skills…," thus making the teams less effective than they could be.

Unfortunately, there are some deficiencies with this investigation; the analysis is completely qualitative and are not quantitatively substantiated. Thomsett presents an interesting study that needs further investigation. He presents general information that

appears very useful but has only intuitive supporting arguments and no empirical evidence. Quantitative data needs to be examined in order to support his ideas.

3. Research Project

This paper is concerned with a single experiment that is part of a much larger research investigation. The fundamental issue that the overall project addresses is how to improve software development teams using knowledge of role theory and personality characteristics. This includes analyzing extant teams and forming new teams. The project consists of four components:

1. Controlled experiments in which college seniors in a software engineering lab are arranged on teams to accept or reject specific hypotheses that test the importance of selected Belbin roles on teams (The first of these experiments is the ultimate topic of this paper.)

2. Analysis of teams from a regional collegiate programming contest to refine analysis techniques for use in the final component of the research, the industry application

3. Analysis of other student programming teams to do a partial validation of the analysis, also with the intent of refining analysis techniques for the industry validation

4. Analysis of industry teams from various companies in order to demonstrate the general utility of the approach

The first two components are complete, data is being gathered for the third, and organizations are being contacted for the final phase. The remainder of this paper presents the controlled experiments, focusing on the analysis and results of the first experiment.

4. The Controlled Experiments

The importance of this research lies in the cause-and-effect relationships that can be used to improve teams extant teams or to form good teams. The effects are observed when some teams are specifically designed to be better than others in terms of the focus of this research: performance and viability. The performance measure for these experiments is time-to-completion. Thus, this part of the research demonstrates that teams that contain specified roles perform better, i.e. finish sooner on the average, than teams that do not possess those roles.

> "Cause-and-effect relationships can be inferred from randomized experiments but not from observational studies. The problem with observational studies is that confounding variables – identifiable or not – my be responsible for observed differences." [Rams97]

This necessitates controlled experiments, making other considerations equivalent and blocking the treatments, teams in this case, so that each is considered equal except for this item being investigated by the experiment. This is the basic difference between controlled experiments and observation studies [Ottr93].

The specific purpose of these experiments is to produce empirical results to support some of the concepts central to the research. Several hypotheses were proposed to focus the research on issues that could be directly tested. Each of the hypotheses that were ultimately selected involves investigating the importance of a team containing one of the Belbin roles. Specifically, the hypotheses are:

Experiment 1: Leadership

H_0: Teams containing one and only one "leader" **do not** do better than teams with no leader or multiple leaders

Experiment 2: Plants

H_0: Teams composed of all Plant team members **do not** perform better than teams with no Plants or even a single Plant.

Experiment 3: Monitor Evaluator

H_0: Teams composed of all Monitor-Evaluator team members **do not** perform better than teams that have no Monitor-Evaluator.

4.1 Establishing Teams and Groups

To understand how the experiments are conducted, the terms *team* and *group* must be identified as distinct. Each team is composed of three members who have certain role characteristics under observation. Each team is placed in a group that is being used to test the hypotheses in question. Therefore, each team is established to be a "successful" team or a "unsuccessful" team in terms of how it is expected to perform in the experiment. For example, to test the first hypothesis, some teams are established with a single, identifiable leader and are in the "successful" group; other teams are established with no leaders or all leaders and are part of the "unsuccessful" group. The teams and groups are re-formed for each of the three experiments.

Using a table like Table 4.1.1 but including all of the teams, the participants are formed into teams and groups. Approximately half of the teams are placed in a group expected to be successful, i.e. quicker to complete a programming problem; the other

teams are formed with the anticipation that they will do statistically worse than the successful teams in terms of the mean completion time of the group.

For each experiment, the role under scrutiny is made unequal between the groups. For example, all of the teams in one group would have a role and all of the teams in the other group would be deficient in that role. The other roles were made as equivalent as possible and the team members' self-rated experience was balanced between the teams.

The table used to accomplish this team formation consists of the following information. Each row under a team, e.g. Team X, contains the information for one team member. The first column simply lists the teams' names; the second column designates the members' numbers; the subsequent columns represent the members' data for the Belbin roles Chairman (CH), Shaper (SH), Plant (PL), Resource Investigator (RI), Monitor-Evaluator (ME), Company Worker (CW), Team Worker (TW), and Completer-Finisher (CF); and the final column shows the members' experience, which is used to make the teams equivalent.

Table 4.1.1: Example of Data for a Single Team

TEAM	Member	CH	SH	PL	RI	ME	CW	TW	CF		Exp
Team X											
	1	c3	-	-	-	-	W2	-	F9		5
	2	-	**s4**	p2	-	m0	-	-	-		4
	3	-	s3	p7	-	m3	w0	-	-		4

The data in each cell in the table represents that member's score on the Belbin Self-Perception Inventory that indicates the Belbin roles. Cells with a dash (-) in them indicate that the score was not significantly high enough to take into account when

forming the teams. You should note that if an individual took the test to distribute the scores as evenly as possible, each cell would have the average of 8.75, therefore a score of 9 or less hardly seems significant. Each cell contains a letter and a digit. The letter represents the role, which is also shown by the column headers abbreviations, and the digit indicates the score on the Belbin test. Scores in the range 10 to 19 are indicated with a lower case letter for the role, e.g. c8 and p2 indicate a Chairman score of 18 and a Plant score of 12; score of 20 and above are indicated with a capital letter, e.g. R2 and M0 indicate a Resource Investigator score of 22 and a Monitor-Evaluator score of 20, which are very high scores. This scheme is used because it removes a lot of extraneous information so that the enormous amount of data that is needed to form the teams evenly, except for the variables under study, is comprehendible.

You should also note that an individual can fill more than one role but also only fulfills a limited number of roles and that, although an individual might have a number that appears to make them fill that role, they may not fill that role because they have a stronger tendency to fill other roles. For example, in Table 4.1.1, Member 1 is a very strong Company Worker (indicated by the W2, as opposed to a simple w2) and a very strong Completer-Finisher. These two roles are so strong that the Chairman role might actually not be filled by this individual. The third member of this team also has a dominate role, the Plant, and secondary roles, Shaper and Monitor-Evaluator; the fourth role, Company Worker, is so much weaker so as not be considered at all. This also brings up the point made by Belbin that if one team member is significantly stronger in a role than another,

then the stronger member tends to dominate, and the weaker to focus more on their other roles, particularly when they have other roles that are stronger than the role in question, e.g. Shaper for Member 3.

4.2 Conducting the Experiments

The teams and groups were formed prior to the commencement of the experiment, and the participants showed up in a laboratory where there are nine computers used in the experiment. There were twenty-four participants who had been formed into eight teams; one computer was used by the experimenter to accept and to test problem solutions. Each participant was given a copy of the problem to be solved that day. The problems were intended to be solved in a little over an hour, although some teams ended up requiring up to two hours. In order to make everything as balanced as possible among the teams, each team could only work on one workstation and were told that they could use whatever editor, compiler, or utilities they wish. The systems being used were running Free BSD 1.1.6. All of the facilities on the machines were equivalent. They could use any of the compilers available for C or C++. Teams emailed their solutions to the experimenter as soon as they felt that they had a working solution, and the experimenter tested the solutions with his own test data. A completion time was only recorded once a submitted solution was satisfactorily tested. Although the problem statements included some test data, the teams were informed that they must create test data themselves.

Each of the experiments involved four problems over a two week period, two lab periods per week. The series of three experiments required a total of twelve weeks. At

the end of the last session for each experiment, the participants were asked to fill out a questionnaire that addresses team viability issues.

4.3 Quantitative Analysis

The analysis of the controlled experiments has two parts that correspond to the two types of data collected for each experiment: performance and viability. The primary interest is in the performance of the groups, viz. the mean time-to-completion. Some objective, quantitative measure needs to be used to provide a "measure of success." In this research, that measure is the time to complete a programming problem. The mean completion times of the g*roups* are compared to determine whether there is a statistical difference between their means. One could certainly argue that this is a small measure of success, but this measure was chosen because it is objective, quantitative, and easy to collect.

The basic hypothesis is that the mean of the completion times for the teams in the successful, "good" group in each of the experiments is significantly lower than for the unsuccessful, "bad" group. In other words, on the average unsuccessful teams should take longer to solve the problems.

Due to space constraints, the viability analysis is not included. The fundamental issue of interest with the viability analysis concerns whether the "good" teams are more viable than the "bad" ones. For example, how good was the inter-member coordination of your team? ... the cooperation? ... the cohesion? How well was each members' technical abilities used? *How willing would you be to work with this team again?* Each member is

asked to rate their team on each of these issues on a scale of one to six. This data is analyzed to determine if one can distinguish between the groups.

Experiment 1: Leadership

This experiment is concerned with leadership on a team, and the null hypothesis under test and attendant alternative hypothesis are:

H_0: Teams containing one and only one "leader" **do not** do better than teams with no leader or multiple leaders.

H_1: Teams containing one and only one "leader" do better than teams with no leader or multiple leaders, as indicated primarily through the Shaper role.

The experiment demonstrates that teams with a single, recognized leader do better than those with no leader. "Successful" teams were established with a single, recognized leader on each team and as many as possible of the other factors roughly equivalent; the "unsuccessful" teams consisted of teams that either had **no** leader type present or had no **clear leader**, i.e. 2 or all 3 were leader types, which caused conflicts of who was in charge and which "direction" the team should pursue. The teams were blocked by level of programming experience. See Table 3. The two Belbin roles Chairman and Shaper are both types of leaders; they *can* be complementary roles, where the Chairman observes the team, knows the strengths and weaknesses of each member (and hence the team), and knows how to take maximal advantage of the team's human resources. The Shaper plays more of an active part in the team, pushing the team members by questioning opinions and decisions, making members take extra effort in their work because a Shaper is very motivating in a confrontational way. Chairmen tend to be introverted and thoughtful; Shapers tend to be extroverted and motivating through making members extend their

capabilities. "The 'Chairman' of (three-person) team(s) would be something of a misnomer" [Belb81] and does not have a significant impact on the team because coordinating the resources of two subordinates is not significant. Therefore, this experiment focuses on the Shaper types of leaders; further cause for ignoring Chairman leaders is the virtual dearth of Chairman types in the class sample.

Table 4.3.1: Teams Used in Experiment 1: Leadership

TEAM	Member	CH	SH	PL	RI	ME	CW	TW	CF		Exp
Team D											
	1	c3	-	-	-	-	W2	-	f9		5
	2	-	**s4**	p2	-	m0	-	-	-		4
	3	-	s3	p7	-	m3	w0	-	-		4
Team F											
	1	-	**s1**	-	-	m4	-	-	-		5
	2	**c0**	-	p0	-	-	w0	t4	-		5
	3	-	-	-	-	M0	W5	T0	-		2
Team H											
	1	-	**s1**	P0	r2	-	-	-	-		4
	2	-	-	P0	-	-	w1	-	-		4
	3	-	-	-	-	m1	w1	t8	-		4
Team G											
	1	**c0**	-	-	-	**m4**	**w1**	-	-		4
	2	-	-	p0	-	-	w4	t3	f0		3
	3	-	-	-	-	-	w9	t5	-		6
Team E											
	1	-	s0	**p3**	-	-	**w1**	**t6**	-		2
	2	-	-	-	-	-	w4	-	f1		5
	3	-	-	p3	r0	m1	w4	-	-		5
Team C											
	1	-	**s2**	-	-	-	w2	-	-		3
	2	c0	**s4**	-	-	-	w1	-	f2		6
	3	**c4**	-	-	-	-	w5	t4	f2		5
Team B											
	1	-	**s2**	-	r2	-	w0	-	-		5
	2	-	**s0**	p3	-	-	w4	-	-		5
	3	c0	**s2**	-	r2	m2	-	-	-		3
Team A											
	1	-	**s0**	P2	-	-	-	-	f5		5
	2	-	**s3**	-	-	m6	w7	-	f0		6
	3	-	**s9**	-	-	m3	w3	-	-		2

After transforming the data with a ln() function, the data showed a significant difference between the groups: group 1 being the teams with a leader and group 2 being teams without a leader. An ANOVA using a SAS program shows that the means of the two groups, 101.81 and 76.63 minutes, are statistically different with a p value of 0.0068. This allows the null hypothesis to be rejected and the alternate hypothesis to be accepted,

which is interpreted as the teams with a single, recognized leader are statistically different (in this case in a better way) than the teams with no clear leader.

Teams D, F, and H (above the bold line in Table 4.3.1) contain a single leader, who has their leader role in bold. These teams form the "successful" group. Teams A, B, and C contain multiple leader roles, who tend to introduce conflict into the teams composition, reducing the effectiveness of the team. Teams E and G have no obvious leader who can be identified by Belbin's roles, although as described below, the c0 member of Team G did indeed act as a leader.

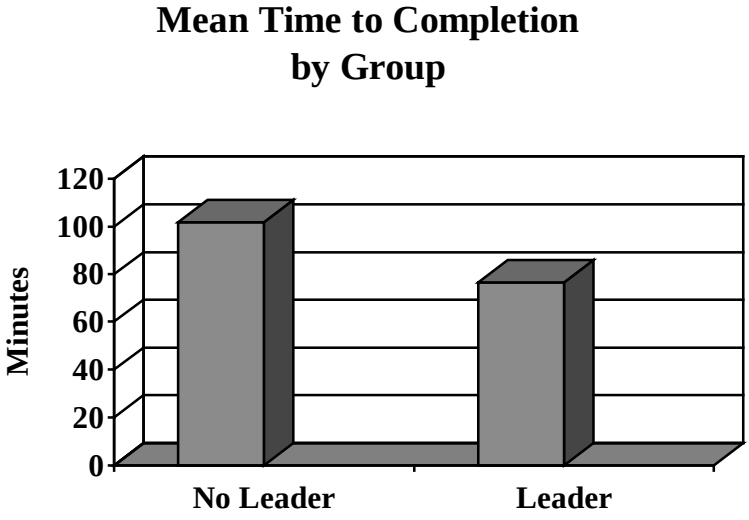

Figure 4.3.1: Differences Between the Means of the Groups – Analysis 1

The above output was generated using Teams A, B, C, E, and G as "unsuccessful" teams. These include both the no-leader teams as well as the conflicting-teams. As it turns out, the individual on team G with a Chairman score of 10, i.e. a c0, acted as the

leader of the team. This can be explained, post hoc, by the fact that the individual having a degree of the leadership role inherent and that the other two roles that the individual should have filled were not sufficiently stronger than the Chairman role, and the Chairman role needed to be filled. Typically when examining the roles to setup the team, only the top two roles were used for each individual. Below is an analysis of the data putting Team G in the "successful" group, which had a single leader, although it was not obvious from the Belbin data.

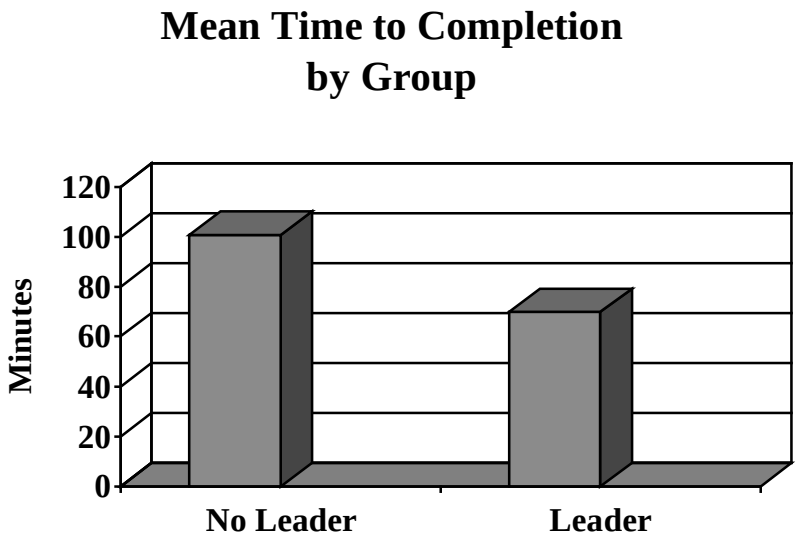

Figure 4.3.2: Differences Between the Means of the Groups – Analysis 2

Figure 4.3.2 shows the resulting means when analyzing the data using Teams A, B, C, and E as the "unsuccessful" teams separates the groups even more. This analysis has better results, as shown by a *p* value of 0.0034 in an ANOVA, with means of 100.90 and 69.75 minutes.

Many subjective observations that were noted both during and immediately after each class period support these empirical results by the experimenter: "The team obviously works well together." "One person (is) at the helm, with the other two supporting technically, non-passively; the leader listens fairly well but is not the bright one." "Passive-aggressive leader is bright and doesn't listen well but listens well enough to get the good ideas from the other members; he tends to become almost too focused sometimes." By the fourth (and last) problem for this experiment, Team F said they "had it down" in terms of coordinating together. Some comments from and about the "unsuccessful" teams include: "...very separated group," "Two dominate individuals with the third being very passive; one member takes the leadership role but only to argue with the other dominate member over everything, although the leader did make attempts to listen to the quite one." By the fourth (and last) problem Team E had "completely given up, assuming that their team had been 'set up' not to work well together." This indicated that the class had come to the conclusion that since we were experimenting with teams, some teams must be set up to perform well and other to perform poorly, which was correct.

5. Conclusions and Continuing Work

Obviously, the conclusion of this study is the alternate hypothesis stated above: Teams containing one and only one "leader" do better than teams with no leader or multiple leaders, as indicated primarily through the Shaper role. Intuitively, this appears obvious, however, controlled experiments are necessary to support the idea. Also,

controlled experiments allow us to apply cause-and-effect to the equation. As this research continues, more definitive statements will be verified. This research provides guidance to managers in forming successful teams as well as in evaluating teams to identify deficiencies in extant teams, with the intent of rectifying such deficiencies.

As outlined in Section 3, this small experiment is not only one of three controlled experiments, but it is just part of a much larger research investigation that is ongoing. Work continues on analyzing the programming competition data and the data from the additional student teams, further refining how to apply Belbin's roles to small software teams. The final phase of the investigation, Analyzing Industry Teams, is just beginning, and we are still interested in getting as many teams as possible to participate.

Although some managers *might* use the personality information directly in that they want to hire "computer science" types and although employee selection based on this is *possible*, this research shows that team improvement can be better accomplished by combining personality information with the concepts of roles and the need to have diverse teams. Team roles need to be the focus of this information, and the roles need to be filled with the appropriate individual. If teams are properly formed using roles, then the team members are happier and more likely to remain with an employer, thus increasing the viability of the team. All in all, fulfilling the goals of this research will produce a model that can be used to improve both the performance and the viability of teams and, therefore, the effectiveness.

Bibliography

[Belb81] Belbin, R. Meredith, Management Teams, John Wiley & Sons, New York, 1981.

[Belb93] Belbin, Meredith, Team Roles at Work, Butterworth-Heinemann, Ltd, Oxford, 1993.

[Bidd79] Biddle, Bruce J., Role Theory: Expectations, Identities, and Behaviors, Academic Press, New York, 1979.

[Boeh81] Boehm, Barry W., Software Engineering Economics, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1981.

[Bost81] Bostrom, Robert P. and Kate M. Kaiser, "Personality Differences within Systems Project Teams: Implications for Designing Solving Centers," *Proceedings of the Eighteenth Annual Computer Personnel Research Conference*, June 3-5, 1981, Association for Computing Machinery Press, New York, pp. 248-285.

[Broo87] Brooks, Fred P., Jr., "No Silver Bullet - Essence and Accidents of Software Engineering," *IEEE Computer*, April, 1987.

[Cher81] Cherlin, Merrill, "The Toll of Turnover," *Datamation*, April, 1981.

[Cont86] Conte, Samuel D. , H. E. Dunsmore, and Vincent Y. Shen, Software Engineering Metrics and Models, The Benjamin/Cummings Publishing Company, Menlo Park, California, 1986.

[Cook93] Cook, Curtis R., Jean C. Scholtz, and James C. Spohrer, Empirical Studies of Programmers: Fifth WorkShop, Dec 3-5, 1993, Palo Alto, CA, Ablex Publishing Corporation, Norwood, NJ, 1993.

[Curt86] Curtis, Bill, Elliot M. Soloway, Ruven E. Brooks, John B. Black, Kate Ehrlich, and H. Rudy Ramsey, "Software Psychology: The Need for an Interdisciplinary Program," *Proceedings Of the IEEE*, August, 1986.

[Curt90] Curtis, Bill, "Empirical Studies of the Software Design Process," *Proceedings of INTERACT90*, August 27-31, 1990, Cambridge, U. K., North-Holland, pp. xxxv-xl.

[DeMa87] DeMarco, Tom and Timothy Lister, Peopleware: Productive Projects and Teams, Dorset House Publishing Co., New York, 1987.

[Flor91] Flor, Nick V., "Analyzing Distributed Cognition in Software Teams: A Case Study Of Team Programming During Perfective Software Maintenance," *Empirical Studies of Programmers: Fourth Workshop*, New Brunswick, New Jersey, December 7-9, 1991, eds. Jurgen Koenemann-Belliveau, Thomas G. Mother, and Scott P. Robertson, Ablex Publishing Corporation, Norwood, New Jersey, pp. 36-59.

[Henr83] Henry, Sallie M., "A Project Oriented Course on Software Engineering," Proceedings of the 1983 ACM SIGCSE Symposium on Computer Science Education, February, 1983, pp. 57-61.

[Henr93] Henry, Sallie and Matthew Humphrey, "Comparison of an Object Oriented Programming Language to a Procedural Language for Effectiveness in Program Maintenance," *Journal of Object Oriented Programming*, Vol. 6, No. 3, 1993.

[Holt87] Holt, Robert W., Deborah A. Boehm-Davis, and Alan C. Schultz, "Mental Representations of Programs for Student and Professional Programmers," *Empirical Studies of Programmers: Second Workshop*, Washington, D.C., December 7-8, 1987, eds. Gary M. Olson, Sylvia Sheppard, and Elliot Soloway, Ablex Publishing Corporation, Norwood, New Jersey, pp. 33-46.

[Hump89] Humphrey, Watts S., Managing the Software Process, Addison-Wesley Publishing Company, Reading, Massachusetts, 1989.

[Hump95] Humphrey, Watts S., A Discipline for Software Engineering, Addison-Wesley Publishing Company, Reading, Massachusetts, 1995.

[Jeff87] Jeffery, D. R., "The Relationship Between Team Size, Experience, and Attitudes and Software Development Productivity," *Proceedings of COMPSAC87*, Tokyo, Japan, October 7-9, 1987, IEEE Computer Society Press, pp. 2-8.

[Know82] Knowles, Eric S., "From Individuals to Group Members: A Dialectic for the Social Sciences," in Personality, Roles and Social Behavior, Eds. William Ickes and Eric S. Knowles, Springer-Verlag, New York, 1982.

[Koen91] Koenemann-Belliveau, Jurgen, Thomas G. Hoher, and Scott P. Robertson, Empirical Studies of Programmers: Fourth WorkShop, New Brunswick, New Jersey, Ablex Publishing Corporation, Norwood, New Jersey, 1991.

[Lewi92] Lewis, John A., Sallie M. Henry, Dennis G. Kafura, and Robert S. Schulman, "On the Relationship Between the Object-Oriented Paradigm and Software Reuse: An Emipirical Investigation," *Journal of Object Oriented Programming*, July-August, 1992.

[Mill83] Mills, Harlan D., Software Productivity, Little, Brown and Company, Boston, 1983.

[Myer80] Myers, Isabel Briggs, Gifts Differing, Consulting Psychologists Press, Inc., Palo Alto, California, 1980.

[Myer85] Myers, Isabel Briggs and Mary H. McCaulley, A Guide to the Development and Use of the Myers-Briggs Type Indicator, Consulting Psychologists Press, Inc., Palo Alto, California, 1985.

[Myer87] Myers, Isabel Briggs, Introduction to Type, Consulting Psychologists Press, Inc., Palo Alto, California, 1987.

[Nels91] Nelson, R. Ryan, Michael W. Kattan, and Paul H. Cheney, "Training, Ability, and The Acceptance of Information Technology: An Empirical Study of IS Personnel and End Users," *Computer Personnel*, November, 1991.

[Nich82] Nichols, Jean A., "Forward," in *Proceedings of Human Factors in Computer Systems*, March 15-17, 1982, Gaithersburg, Maryland, Association for Computing Machinery Press, p. iii.

[Olso87] Olson, Gary M., Sylvia Sheppard, and Elliot Soloway, Empirical Studies of Programmers: Second WorkShop, Ablex Publishing Corporation, Norwood, New Jersey, 1987.

[Ottr93] Orr, R. Lyman, An Introduction to Statistical Methods and Data Analysis, Duxbury Press, Belmont, California, 1993.

[Paul95] Paulk, Mark C., Charles V. Weber, Bill Curtis, and Mary Beth Chrissis, The Capability Maturity Model: Guidelines for Improving the Software Process, Addison-Wesley Publishing Company, Reading, Massachusetts, 1995.

[Rams97] Ramsey, Fred L. and Daniel W. Schafer, The Statistical Sleuth: A Course in Methods of Data Analysis, Duxbury Press, Belmont, California, 1997.

[Sarb54] Sarbin, Theodore R., "Role Theory," in Handbook of Social Psychology, Ed. Gardner LindzeyAddison-Wesley Publishing Company, Reading, Massachusetts, 1954.

[Scot95] Scott, Thomas J. and James H. Cross II, "Team Selection Methods For Student Programming Teams," *Software Engineering Education: Proceedings of 8th SEI CSEE Conference*, New Orleans, Louisiana, March 29 - April 1, 1995, Springer-Verlag, pp. 295-303.

[Shne80] Shneiderman, Ben, Software Psychology, Winthrop Publishers, Inc., Cambridge, Massachusetts, 1980.

[Solo86] Soloway, Elliot and Sitharama Iyengar, Empirical Studies of Programmers, June 5-6, 1986, Washington, D.C., Ablex Publishing Corporation, Norwood, New Jersey, 1986.

[Stra94] Straus, Susan G. and Joseph E. McGrath, "Does the Medium Matter? the Interaction of Task Type and Technology on Group Performance and Member Reactions," *Journal of Applied Psychology*, Vol. 79, No. 1, 1994.

[Sund90] Sundstrom, Eric, Kenneth P. De Msuse, and David Futrell, "Work Teams," *American Psychologist*, February, 1990.

[Thom90] Thomsett, Rob, "Effective Project Teams," *American Programmer*, July/August 1990.

[Trow91] Trower, Jonathan K. and Detmar W. Straub, Jr., "Improving the Performance of Technologists and Users on Interdisciplinary Teams: An Analysis of Information Systems Project Teams," *Computer Personnel*, November, 1991.

[vonM84] von Mayrhauser, Anneliese, "Task Analysis and the Selection of Software Development Team Structures," *Proceedings of COMPSAC84*, Chicago, Illinois, November 7-9, 1984, IEEE Computer Society Press, pp.120-126.

[Walz87] Walz, Diane B., Joyce J. Elam, Herb Krasner, and Bill Curtis, "A Methodology for Studying Software Design Teams: An Investigation of Conflict Behaviors in the Requirements Definition Phase," *Empirical Studies of Programmers: Second Workshop*, Washington, D.C., December 7-8, 1987, eds. Gary M. Olson, Sylvia Sheppard, and Elliot Soloway, Ablex Publishing Corporation, Norwood, New Jersey, pp. 83-99.

[Wein71] Weinberg, Gerald M., The Psychology of Computer Programming, Van Nostrand Reinhold Company, New York, 1971.

[Zahn90] Zahniser, Richard A., "Building Software in Groups," *American Programmer*, July/August 1990.

Appendix A: Description of Belbin Roles

A **Chairman** is an individual who controls the team in a typical head-of-the-table manner. This role guides the team towards what he or she perceives as the teams objectives in the best manner he or she can determine. A Chairman is calm and listens to other team members well, having a natural ability to get the most and best out of everyone's potential. Additionally, he or she is very aware of the strengths and weaknesses of the team as a whole. The Chairman is one type of team leader.

The second role, **Shaper**, is also a leader type but has a completely different personality and managerial style than a Chairman. A Shaper is a slave driver, questioning members to find the best approaches to problems. This role leads the team by stimulating the members to "challenge inertia, ineffectiveness, and complacency." Shapers tend to be nervous, extroverted, competitive, and argumentative, just to name a few of their stronger characteristics.

The question arises of who is leader of the team if individuals of both roles are present in a team. The answer is that either they cooperate in their leadership, trading off at designated appropriate times, or they produce a great deal of conflict in the team and can make the team less successful. Ideally, both roles are present on a team, and they complement each other. The Chairman of the team tries to keep progress moving in a steady, conservative fashion, and the Shaper likes to rile things up to make sure that all

possibilities are being considered. For each of the eight roles, a similar relationship exists - there is a complement of some sort for each role.

The third role is the **Plant**, mentioned in the introduction of this section. The Plant is the innovator of the team; he or she advances new approaches and ideas with special attention to major issues. A Plant is typically introverted, unorthodox, imaginative, and intelligent but "inclined to disregard practical details or protocols." The Plants are the brainchildren who must be nurtured and occasionally drawn back into the real world because they tend to have their heads in the clouds. They are considered one of the intellectual types in a team.

The complementary role to the Plant is the **Monitor-Evaluator**, the other intellectual. The Monitor-Evaluator is the analyzer of the team; the member who evaluates all of the alternatives for all decisions, methods, and approaches so that the team is positioned as competitively as possible. This role can be under-appreciated because members that fill this role tend to be unemotional, dry, over-critical, hard-headed, and unmotivational to others; this is terribly unfortunate because the role is necessary for the team to succeed. The Monitor-Evaluator plays counterbalance to the Plant: Plants come up with pie-eyed ideas and the Monitor-Evaluator is the only type of role that can debate successfully with a Plant to make correct choices.

The fifth role is **Resource Investigator**, who is considered one of the negotiator types of roles. In one aspect, Resource Investigators are very similar to Plants, although not considered complements. Both of these roles are seen as creative members who bring

innovation to the team. The difference is that Resource Investigators get their innovations from sources external to the team, primarily due to their extroverted nature. Other attributes associated with this role are enthusiasm, curiosity, and communication skills. These members also have a tremendous "capacity for contacting people and exploring anything new;" unfortunately, they tend to lose interest in a problem or situation once the novelty of it wears off. Their strength lies in that they know how to get what they want out of a person.

The complementary negotiator role is the **Team Worker**, who makes sure that the team works together towards their goals. Whereas the Resource Investigator negotiates outside of the team to get what the team needs, the Team Worker facilitates or negotiates within the team. For example, no matter how brilliant a Plant may be or how successful a team leader may be, team members need to get along in order for the team to be effective or successful. Further, members may get irritated by the slow, thoughtful decision-making process of the Monitor-Evaluator; members may not let the introverted Plant express herself or himself. Therefore, some member role is necessary to make sure that everyone gets along and is provided the opportunity to accomplish their functions. Team Workers tend to be very socially oriented, mild, and sensitive; they have a strong ability to respond, communicate, and deal with people and situations, the ultimate facilitator.

The Seventh role that Belbin discusses is the **Company Worker**, who is the meat-and-potatoes member of the team. Company Workers' primary function is "turning concepts and plans into practical working procedures;" they focus on "carrying out

agreed (upon) plans systematically and efficiently." Company Workers tend to be conservative, dutiful, predictable, good organizers, hard-working, and self-disciplined. On the negative side, they are inflexible and resistant to unproven ideas, views, and approaches; they support the status quo a bit extremely. Although Company Workers tend to exhibit some negative aspects, one should bear in mind that negative aspects are part of what defines the individual who can fulfill the role successfully.

The final role, **Completer-Finisher**, is the complement of the Company Worker in a manager-worker relation. Completer-Finishers focus on a project delivering on-time and within cost estimates; the Company Worker carries out work to accomplish this. Primarily, team members in this role focus on avoiding "mistakes of both omission and commission, actively searching for aspects of work (that) need a more than usual degree of attention." They stress a sense of urgency to the team in order to meet deadlines and goals. Team members who fill this role are painstaking, orderly, conscientious, and anxious; they demonstrate a strong sense of follow-through and perfectionism.