2005

Collaborative Software Development in Agile Methodologies - Perspectives from Small Group Research

VenuGopal Balijepally
University of Texas at Arlington, venugopal@uta.edu

Collaborative Software Development in Agile Methodologies – Perspectives from Small Group Research

VenuGopal Balijepally
University of Texas at Arlington
venugopal@uta.edu

ABSTRACT

Collaborative software development is a hallmark of agile methodologies such as Extreme Programming (XP). These methodologies have practices like pair programming, where two programmers collaboratively work on all aspects of software development. There is however a dearth of empirical research in this area. Studies with sound theoretical underpinnings and strong empirical rigor are called for to inform the software practice of the effectiveness of this important practice. While such collaborative working is relatively new to software community, small group research has grappled with it over the years looking for the task and other contingencies impacting the effectiveness of collaborative working versus individual working. In this paper, we provide a brief overview of some research streams in small group research and social psychology that could potentially inform IS research on collaborative software development. These small group research topics include group task typologies, individual versus group performance, social facilitation, social loafing, and group motivational gains. We then discuss implications for research on collaborative programming and provide some illustrative research questions.

Keywords

Pair programming, agile development, problem-solving, group performance, social motivation

INTRODUCTION

Software development has been undergoing rapid transformation with new methodologies and approaches emerging to address the limitations of existing software practice and to cater to the changing business requirements. The need for flexibility and adaptation in the face of shifting business and technical landscapes is one of the factors contributing to the popularity of agile methodologies. The increasing coverage of agile methodologies in the practitioner literature is suggestive of the fact that agile development is here to stay.

Extreme Programming (XP) pioneered by Kent Beck and colleagues is one of the better documented and more popular of the agile methodologies. One of the most important, yet controversial of the XP practices is pair programming. Pair programming involves two programmers collaboratively working on all aspects of the software development. XP proponents claim that the benefits of pairing such as improved software quality, enhanced motivation and satisfaction of developers, knowledge transfer and adaptive learning are well worth the additional costs involved. The empirical evidence is however limited. This area is in need of research studies with sound theoretical underpinnings and strong empirical rigor to inform the software practice of its effectiveness. However the interpersonal dynamics involved when two programmers work collaboratively together are highly complex. Small group research could be a logical reference domain for this area to explore the social and behavioral aspects of collaborative working.

When two programmers are put together to work on a programming task that is traditionally performed by autonomous individuals, it highlights a fundamental question of whether groups are better than individuals on certain tasks? This has been a classic research question that captivated social psychologists and organizational researchers over the years. Though there have been no definitive answers, one consensual finding of this stream of research is that the individual versus group effectiveness is highly dependent upon the task at hand (Hill, 1982). Based on this realization, the early social psychologists have developed several group task classifications. Lot of theoretical and empirical literature is also available in the areas of individual versus group effectiveness, social facilitation, and group motivation. For studying collaborative programming as in XP, understanding these group task typologies would help IS researchers in framing the software tasks in terms of these group tasks. This will help bridge the two literatures and allow relevant research findings to be drawn. An understanding of the group literature should also help identify various contextual and task related factors that potentially impact the processes and dynamics of collaborative group work. In this paper we discuss the group task typologies and some theoretical perspectives and empirical evidence from small group research, that could potentially inform IS research in the area of team-based software development in general and pair programming in particular.

The organization of the paper is done as follows: First we provide a brief overview of agile philosophy and pair programming followed by a review of group task typologies. Second we provide a summary of key findings and literatures on individual versus group effectiveness, social facilitation, social loafing, and group motivational gains. Third we discuss the issues and challenges involved in drawing findings from these research areas for studying collaborative programming in XP. Some illustrative research questions on collaborative programming are suggested that benefit from the theoretical perspectives presented here. Finally, we discuss conclusions.

AGILE METHODOLOGIES AND PAIR PROGRAMMING

The agile methodologies place a huge premium on people and interactions over any other technical or process related factors. For example, XP stipulates small programmer teams of 3 to 10 members with a collocated customer providing ongoing user perspective and feedback. XP also has pair programming as a core practice, wherein two programmers work together at the computer terminal taking turns with the keyboard. The programmer holding the keyboard does coding and acts as the driver. The partner acting as the navigator, actively inspects the code, looks for errors, and thinks strategically about the program logic. The XP advocates insist that pair programming results in higher quality code, higher satisfaction among the developers and reduced overall costs due to a higher quality finished product (Beck, 2000). With agile methodologies gaining increasing attention and acceptance, it is imperative that practices such as pair programming and the collaborative processes inherent therein need research attention from the IS community.

GROUP TASK TYPOLOGIES

Programming in general may be considered as a problem-solving activity as there usually is a correct answer, though it may not be very compelling. In this section a brief overview is provided of the group task typologies particularly relevant to problem solving tasks. An understanding of group task typologies should help frame software tasks in terms of group tasks, so that relevant findings could be drawn from group research.

Steiner proposed a typology based on three issues – task divisibility, importance of quality vs. quantity, and method of combining group inputs. Based on divisibility, the tasks may be considered as divisible (subtasks exist) vs. unitary (no subtasks exist). In terms of the importance of quantity vs. quality the tasks are categorized as maximizing vs. optimizing (Steiner, 1972). The tasks are further classified based on how members' efforts are combined to yield the group product. Disjunctive tasks involve selecting from individual judgments. They are typically unitary (not divisible) and optimizing (quality is emphasized). The group discusses till its members agree on a solution such as in juries and problem-solving work teams. In disjunctive tasks, the group performance is driven by the talents and knowledge of the best group member. In conjunctive tasks as in assembly line, the groups' performance is limited by the worst performing member. Additive tasks involve combining the group members' contributions such as when a group paints a house. The performance in additive tasks is dependent upon the abilities of 'average' group member. In discretionary tasks, the group decides the way to organize inputs as in self-managed teams (Steiner, 1972).

Laughlin articulated a group task continuum anchored by *intellective* and *judgmental* tasks. *Intellective* tasks have a demonstrably correct answer, while *judgmental* tasks are evaluative, behavioral, or aesthetic judgments with no correct answers (Laughlin and Ellis, 1986). McGrath proposed a task circumplex in which the vertical axis denotes the degree to which the task involves collaboration and coordination or conflict resolution. The horizontal axis indicates the degree to which the task entails cognitive or behavioral performance. The circumplex contains eight different task types. Intellectual, judgmental, and idea generation tasks that fall in the cognitive end of this spectrum are described below (McGrath, 1984).

Intellective tasks or problem solving tasks have demonstrably correct answers, and require choosing correct answers. Consensus is required, but once the solution is recognized there is often little to debate. If anyone in the group does solve the problem, then the group has solved it. Tasks falling in this category include Laughlin's intellective tasks with correct and compelling answers, problem-solving tasks with correct but not compelling answers, and tasks where expert consensus defines answers. The key notion is the correct answer. Decision making tasks or judgment tasks do not have a correct answer, but involve reaching consensus on a preferred answer. Attaining consensus requires communicating not just facts, but also values, beliefs, and attitudes about the merits of alternative solutions as in jury tasks. Idea generation tasks involve creativity and have more than one correct solution as in brainstorming (McGrath, 1984).

Group performance research is organized in terms of these different task types. The nature of software task in a given collaborative context may be analyzed based on the above group task types to understand the expected dynamics and performance implications.

RESEARCH ON PERFORMANCE OF SMALL GROUPS

Group performance in various tasks has been studied in the literatures on individual versus group performance, social facilitation, and group motivational gains. In the next few sections these different research streams are discussed to understand the theoretical underpinnings and key literatures.

Individual versus Group Performance

While programming was traditionally conceived as an individual activity, the new paradigm of agile development views it as a collaborative team based effort. An appreciation of the research on individual versus group performance from small group research should help appreciate the relative strengths of groups and individuals on different tasks.

In several problem-solving situations, the individual could be as effective as the group (Hare, 1995). Productivity of group may be conceived as determined by the most competent member, plus process gains due to 'assembly bonus effects' (resulting from efficient group interaction) minus process losses. Assembly bonus effect is realized when group performance is better than the performance of any individual or any combination of individual member efforts. Such effects are generally modest. A general finding of this research stream is that groups are better than the average individual, but rarely better than the best individual (Hill, 1982).

Group effect emanates from having large number of people to generate ideas, identify objects and remember facts. Based on information processing view, groups potentially outperform individuals in highly intellectual problem-solving tasks with large information processing requirements (Hinsz, Tindale and Vollrath, 1997). Group superiority over individuals in problem solving hinges on the demonstrability of the strategies, operations, and procedures that lead to the problem solution. Four conditions of demonstrability identified in literature are: a) availability of sufficient information; b) group consensus on a conceptual system; c) members suggesting incorrect responses being able to identify correct response when proposed; d) member with correct response having sufficient time, ability and motivation to demonstrate it to other members. With increasing demonstrability, problem solving groups show distinctly superior performance over individuals (Laughlin, Zander, Knievel and Tan, 2003).

In judgment tasks groups report fewer but more accurate facts (Hill, 1982). A more statistical group effect explaining the higher accuracy of groups is that the average of a number of judgments is usually more accurate than that of one individual (Laughlin and Barth, 1981). In decision tasks group may be no better than the best individual member (Miner, 1984). One source of process loss in groups is failing to identify and use the resources of capable group members (Kerr and Tindale, 2004). There are also no general patterns discernible in judgment biases of individuals and groups, with group or individual superiority dependent upon several contingent factors (Kerr, MacCoun and Kramer, 1996).

A robust finding in brainstorming research is that nominal groups with individuals working alone produce more ideas than interacting brainstorming groups. There is attributed to process losses such as production blocking (inability of more than one person to talk or even think at the same time), evaluation apprehension, and convergence on a relatively low standard of performance due to social comparison effect (Mullen and Salas, 1991). Computer mediated brainstorming groups have however been found to be superior to nominal groups due to the elimination of some of the above process losses (Dennis and Valacich, 1993).

The above summary highlights the centrality of task and its characteristics to the group effectiveness. The next subsection discusses the phenomenon of social facilitation and the research findings thereof.

Social Facilitation

Social facilitation is a research area in social psychology, where it is demonstrated that in the presence of another individual, the performance on well learned tasks is facilitated, while performance on novel or more complex tasks is hampered (Aiello and Douthitt, 2001). The simple tasks where social facilitation effect was found include negotiating simple mazes, dressing in familiar clothes, fishing reel winding and copying simple material. Some tasks where presence of others was found to hamper performance include solving difficult anagrams, recognition of novel stimuli, dressing in unfamiliar clothes, and negotiating difficult mazes.

The social facilitation effects are explained in terms of drive (Zajonc, 1980), evaluation apprehension (Bond, 1982; Cottrell, 1972), and cognitive theories (Baron, 1986). According to drive theory, presence of another individual during task performance is drive-inducing. Increased drive facilitates dominant, well learned responses, but hampers non-dominant ones. The enhanced drive in the presence of another individual facilitates simple task performance, while hampering complex task performance (Zajonc, 1980). Evaluation apprehension is another explanation offered for the social facilitation effect. Based on social comparison theory Cottrell proposed that it is not the mere presence, but when individuals are concerned about how others may evaluate them that their drive levels could get elevated (Cottrell, 1972). According to self-presentation theory explanation, people make an attempt to appear competent to others. When working on simple tasks the impression management efforts of the individual could facilitate performance. While working on difficult tasks, the embarrassment from committing mistakes could impair performance (Bond, 1982). Baron proposed a cognitive theory to explain social facilitation and argued that attention conflict in the presence of others could produce drive-like effect on performance. The conflict itself could result from both internal and external factors. Performance may be facilitated up to a point by the distraction, beyond which it starts to deteriorate (Baron, 1986).

In the context of pair programming, there could be social facilitation effect in the presence of the partner due to the various theoretical reasons articulated above. However the nature of partner presence needs to be analyzed to judge whether such an effect is likely to occur.

Social Loafing

Studies in small group performance have found that when individuals work in groups collectively on relatively simple tasks, they exert less effort than when they work individually. Reduced risks of evaluation, opportunity to free ride on others' efforts, and unwillingness to shoulder the work of a capable, free-riding member of the group are some of the psychological mechanisms underlying such behavior. The theoretical perspectives and concepts used to explain social loafing are: social impact, arousal reduction, evaluation potential, dispensability of effort, matching of effort, and self-attention (Karau and Williams, 1993) .

According to social impact theory, individuals in a social situation may be viewed as either as source or targets of social impact. The extent of social impact experienced by an individual is a function of the strength, immediacy, and number of sources and targets of social impact (Latane, 1981). In the group condition of social loafing studies, the experimenter urging the subjects to try as hard as possible may be considered as the source of social impact with the subjects being the targets. The impact of the experimenter is divided among the several target subjects thus resulting in reduced effort with increase in the size of group (Karau and Williams, 1993).

One explanation for social loafing in groups is lack of evaluation of individual output so that members can 'hide in the crowd'. Group members become aware that may not get fair share of credit or blame for group performance. In many situations making individuals' collective inputs verifiable to anyone including oneself may be sufficient to eliminate social loafing. However two requirements need to be satisfied for evaluation to be possible from any source – the individual's output should be known or identifiable and there should be a standard - objective, social or personal, available for comparison (Harkins and Jackson, 1985).

Social loafing may also be attributed to members' feeling that their contributions are not essential to the group performance. This is especially the case in several disjunctive tasks where the group has solved the problem if any one member is able to do it. Also people expect others to slack off in group work and hence reduce their efforts to maintain equity (Karau and Williams, 1993).

Jackson and Williams articulated that working in a group is drive-reducing when other individuals are not sources of social impact but co-targets of social impact. In social facilitation studies the presence of others is considered as drive-inducing as others are the sources of social impact. However, it is shown that reduced drive experienced in group working, while contributing to social loafing on simple tasks may in fact facilitate performance in novel and difficult tasks (Jackson and Williams, 1985).

In programming pairs, social loafing could be an issue. However other contingent factors articulated above need to be examined to judge the magnitude of any such effects.

Group Motivational Gains

People intuitively expect some motivational gains to occur in group work, though a vast majority of studies have reported of motivational losses as in social loafing. The few conditions where group motivational gains are observed are articulated below.

When individuals expect their co-workers to perform poorly on a meaningful task, they may increase their effort in what is called as social compensation effect (Williams and Karau, 1991). When paired with a group member who is believed to exert low effort, group members work harder when the partner has low abilities, but typically loaf when the partner has high abilities (Hart, Bridgett and Karau, 2001).

Otto Kohler demonstrated motivational gains in certain conjunctive tasks where performance of the group is driven by its weakest member. Using a physical endurance task, Kohler showed that when members of the dyad have moderate differences in abilities (not too similar or dissimilar), they performed better as a pair compared to their expected individual performance (Witte, 1989). The motivational gains were found to result mainly from the weaker member (Stroebe, Diehl and Abakoumkin, 1996).

While social compensation effect is attributed to greater effort put in by the more able partner, Kohler effect is credited to the motivational gains of the less able partner. When there is ability discrepancy within a dyad, which of these two effects is likely to result is contingent on the perceived instrumentality of individual effort to the group performance. Social compensation effect is likely to result in an additive task, when the higher ability partner is likely to work harder to compensate for the low ability partner. Kohler effect is likely to occur in a conjunctive task where the contribution of the low ability member holds the key to the group performance (Williams, Harkins and Karau, 2003). In collaborative programming, any of these motivational gains could result depending upon the ability discrepancies and nature of work organization.

The brief overview of research related to social performance presented here highlights the different perspectives that could potentially inform IS research. The key concepts and literatures reviewed here are summarized in Table 1. The next section discusses the implications and challenges involved in drawing from this body of knowledge to research collaborative programming.

IMPLICATION FOR RESEARCH IN PAIR PROGRAMMING

Software development is typically conceived as involving various activities such as systems analysis, design, implementation and testing. When investigating the efficacy of collaborative pairs engaged in such activities, it is useful to frame software activities in terms of the tasks types articulated in small group research. This is a crucial first step to identify and draw relevant research findings. For example a simple programming task involving the translation of design to code may be categorized as unitary, disjunctive and optimizing based on Steiner's typology. The pair typically works in tandem and arrives at a mutually acceptable solution based on a shared understanding of the problem at hand. The task is optimizing as the emphasis is on the quality of solution. The task may be considered as intellective as there is a correct solution, though it may not be very compelling for difficult tasks. Evidence from group research suggests that performance in such tasks is driven by the abilities and knowledge of the best member of the group.

A fine grained analysis of software activities as indicated above could provide insights into the potential group performance in such tasks. But several other software activities may involve multiple task types of say McGrath's typology. For example, a systems design task may involve generating ideas (creativity task), deciding on issues with no correct answers (judgment task), resolving conflicts of interests (mixed motive task) and so forth. The challenge lies in making a fine grained analysis of the software activity of interest in a given study, to understand its inherent basic task composition.

If the various stages of systems development are done iteratively by the pair as in XP, the challenge of identifying the basic group tasks and drawing on relevant group research findings would be truly daunting. As collaborative software development as in pair programming is a new phenomena, there is no previous IS research that attempted such task analysis of activities involved in software development and this should be a fruitful area for IS researchers.

Research Area	Concepts	Key References
Group Task Typologies		
Steiner's task types	Task categorization based on task divisibility, importance of quality vs. quantity, and method of combining group inputs	(Steiner, 1972)
Laughlin's task classification	Intellective and judgment tasks	(Laughlin and Adamopoulos, 1980)
McGrath's task circumplex	Task circumplex categorizing eight group tasks based on level of collaboration/conflict and cognitive/behavioral performance requirements	(McGrath, 1984)
Individual vs. Group Performance		
General Findings	Group superiority in tasks involving high information processing	(Hinsz et al., 1997; Laughlin et al., 2003)
Problem solving tasks	Conditions of demonstrability of group solution that contributes to group superiority	(Laughlin and Ellis, 1986)
	Experimental designs for Individual versus Group Comparison	(Laughlin et al., 2003)
Judgment Tasks	In decision tasks groups no better than the best member	(Miner, 1984)
	Judgment bias in groups versus individuals – no easy answers	(Kerr et al., 1996)
Brainstorming Tasks	Nominal groups superior to brainstorming groups in idea generation	(Mullen and Salas, 1991)
	Computer mediated groups superior to nominal groups	(Dennis and Valacich, 1993)
Reviews	Reviews of group versus individual performance	(Hare, 1995; Hill, 1982)
Social Facilitation		
Drive explanation	Mere presence of other individual is drive inducing and facilitates well learned responses	(Zajonc, 1980)
Evaluation apprehension	The possibility of evaluation is drive inducing and not mere presence	(Bond, 1982; Cottrell, 1972)
Attention conflict	Attention conflict in the presence of others produces drive like effects	(Baron, 1986)
Reviews	Review of Social Facilitation literature	(Aiello and Douthitt, 2001)
Social Loafing		
Social impact theory	The impact of the supervisor is diffused among several subjects	(Latane, 1981)
Evaluation potential explanation	Not having identifiable individual outputs that could be evaluated	(Harkins and Jackson, 1985)
Dispensability of effort	Perception that their personal contributions do not matter to the group output	(Kerr and Brunn, 1983)
Drive reduction in groups	Working in group reduces drive and thus individuals slack off especially in simple tasks but do better on complex tasks	(Jackson and Williams, 1985)
Reviews	Review of Social Loafing literature	(Karau and Williams, 1993)
Group Motivational Gains		
Social Compensation	Higher ability member making extra effort to compensate for the expected low performance of low ability partner	(Williams and Karau, 1991)
Kohler Effect	Motivational gains for low ability member in conjunctive tasks	(Stroebe et al., 1996; Witte, 1989)

Table 1 – Summary of Key Concepts and Literatures on Individual versus Group Performance

As articulated by Laughlin, with increasing solution demonstrability in intellective tasks, the groups could potentially outperform the best individuals (Laughlin et al., 2003). The sub-tasks involved in systems development are expected to have some aspects of intellective tasks, though they may have features of other tasks as well. The level of solution demonstrability of these intellective components, if assessed, would provide a good heuristic to the expected performance of the pair. Detailed task analysis again has to precede any such attempts.

As brought out in the review, there are several theoretical perspectives such as, social facilitation, social loafing, individual versus group performance, and group motivational gains that could potentially inform the research on the performance implications of collaborative programming. The challenge lies in evaluating each of these approaches and identifying the ones likely to be more influential. There may be seemingly contradictory evidence from different research streams that need reconciliation. For example, social facilitation research argues that mere presence of another individual could hamper performance on complex tasks, while facilitating performance on simple tasks. Evidence from research on individual versus group performance suggests that groups are likely to outperform individuals in complex task performance. Such seemingly contradictory findings need to be resolved theoretically before any heuristics are drawn.

Table 2 provides an illustrative summary of some research areas in pair programming that would benefit from the theoretical perspectives presented here. It is pertinent to note that some or all of the theoretical perspectives presented here could inform any research question. The applicability of each of these theoretical effects needs to be examined individually and how the contingent factors influence the predominance of any particular theoretical effect should be theoretically articulated. For example social loafing could be an issue in several pair programming research questions. The contingent effect of other factors that enhance or diminish this effect such as evaluation potential, task meaningfulness, and task complexity should be examined to assess the expected levels of social loafing. In view of the inherent theoretical complexity of the dynamics involved in collaborative programming, each research question needs a fine grained theoretical analysis.

Theoretical Perspective	Relevant Constructs	Illustrative Research Areas in Pair Programming
Social Facilitation	Task difficulty Nature of Partner presence – evaluative versus non-evaluative	Effect of task difficulty on pair programming outcomes Effect of different types of collaboration (fixed roles versus interchanging roles) on pair effectiveness
Social Loafing	Task meaningfulness Task complexity Individual performance evaluation Ability	Effect of group based rewards on pair outcomes Effect of ability discrepancies and task complexity on pair programming outcomes
Individual vs. Group Performance	Solution demonstrability	Effectiveness of pair programming in different types of systems development tasks
Group Motivation Gains		
Social compensation	Ability Expectation of Partner's effort Task meaningfulness	Effect of ability discrepancies on pair performance outcomes over sustained periods of collaborative working
Kohler effect	Ability	Effect of different types of work organization within pairs on performance outcomes

Table 2 – Small Group Research Perspectives and Research Areas in Pair Programming

CONCLUSION

The need to deliver high quality software that meets user expectations, in a timely and cost efficient manner has ushered in new light-weight approaches, collectively called agile methodologies. These methodologies emphasize collaborative software development in small informal teams with collocated customers. Pair programming is an important practice in Extreme Programming, which is one of the most popular of the agile methodologies. While such collaborative working is relatively new to the software community, small group research has grappled over the years with the issues involved in group versus individual working.

This study provides an overview of various research areas in small group research that are relevant to IS community in researching issues related to collaborative software development. Some issues and challenges involved in drawing from this large body of knowledge are also discussed. Some research areas in pair programming are suggested that could benefit from the theoretical perspectives presented here.

REFERENCES

1. Aiello, J.R., and Douthitt, E.A. (2001) Social Facilitation from Triplett to Electronic Performance Monitoring, *Group Dynamics*, 5, 3, 163-180.
2. Baron, R.S. (1986) Distraction-Conflict Theory: Progress and Problems, *Advances in Experimental Social Psychology*, 19, 1-36.
3. Beck, K. (2000) Extreme Programming Explained: Embrace Change, Addison Wesley Longman Inc., Reading, MA.
4. Bond, C.F. (1982) Social Facilitation: A Self-Presentational View, *Journal of Personality and Social Psychology*, 42, 1042-1050.
5. Cottrell, N.B. (1972) Social Facilitation, in C.G. McClintock (Ed.) *Experimental Social Psychology*, Holt, New York, 185-236.
6. Dennis, A.R., and Valacich, J.S. (1993) Computer Brainstorms: More Heads are Better than One, *Journal of Applied Psychology*, 78, 4, 531-537.
7. Hare, A.P. (1995) Individual Versus Group, in A.P. Hare, H.H. Blumberg, M.F. Davies and M.V. Kent (Eds.) *Small Group Research: A Handbook*, Ablex Publishing Corp, Norwood, NJ, 261-270.
8. Harkins, S.G., and Jackson, J.M. (1985) The Role of Evaluation in Eliminating Social Loafing, *Personality and Social Psychology*, 11, 575-584.
9. Hart, J.W., Bridgett, K.J., and Karau, S.J. (2001) Coworker Abiity and Effort as Determinants of Individual Effort on a Collective Task, *Group Dynamics*, 5, 3, 181-190.
10. Hill, G.W. (1982) Group versus Individual Performance: Are N + 1 Heads Better than One? *Psychological Bulletin*, 91, 517-539.
11. Hinsz, V.B., Tindale, R.S., and Vollrath, D.A. (1997) The Emerging Conceptualization of Groups as Information Processors, *Psychological Bulletin*, 121, 1, 43-64.
12. Jackson, J.M., and Williams, K.D. (1985) Social Loafing on Difficult Tasks: Working Collectively can Improve Performance., *Journal of Personality And Social Psychology*, 49, 4, 937-942.
13. Karau, S.J., and Williams, K.D. (1993) Social Loafing: A Meta-Analytic Review and Theoretical Integration, *Journal of Personality and Social Psychology*, 65, 4, 681-706.
14. Kerr, N.L., and Brunn, S. (1983) The Dispensability of Member Effort and Group Motivation Loses: Free Rider Effects, *Journal of Personality and Social Psychology*, 44, 78-94.
15. Kerr, N.L., MacCoun, R.J., and Kramer, G.P. (1996) Bias in Judgment: Comparing Individuals and Groups, *Psychological Review*, 103, 4, 687-719.
16. Kerr, N.L., and Tindale, R.S. (2004) Group Performance and Decision Making, *Annual Review of Psychology*, 55, 1, 623-655.
17. Latane, B. (1981) The Psychology of Social Impact, *American Psychologist*, 36, 343-356.
18. Laughlin, P.R., and Adamopoulos, J. (1980) Social Combination Processes and Individual Learning for Six-Person Cooperative Groups on an Intellective Task, *Journal of Personality and Social Psychology*, 38, 941-947.
19. Laughlin, P.R., and Barth, J.M. (1981) Group-to-Individual and Individual-to-Group Problem-Solving Transfer, *Journal of Personality and Social Psychology*, 41, 1087-1093.
20. Laughlin, P.R., and Ellis, A.L. (1986) Demonstrability and Social Combination Processes on Mathematical Intellective Tasks, *Journal of Experimental Social Psychology*, 22, 177-189.
21. Laughlin, P.R., Zander, M.L., Knievel, E.M., and Tan, T.K. (2003) Groups Perform Better Than the Best Individuals on Letters-to-Numbers Problems: Informative Equations and Effective Strategies, *Journal of Personality and Social Psychology*, 85, 4, 684-694.

22. McGrath, J.E. (1984) Groups: Interaction and Performance, Prentice-Hall, Inc., Englewood Cliffs, N. J.
23. Miner, F.C. (1984) Group versus Individual Decision Making: An Investigation of Performance Measures, Decision Strategies, and Process Losses/Gains, *Organizational Behavior and Human Performance*, 33, 112-124.
24. Mullen, B., and Salas, J.C. (1991) Productivity Loss in Brainstorming Groups: A Meta-Analytic Integration, *Basic Applied Social Psychology*, 12, 1, 3-23.
25. Steiner, I.D. (1972) Group Process and Productivity, Academic Press, New York.
26. Stroebe, W., Diehl, M., and Abakoumkin, G. (1996) Social Compensation and the Kohler Effect: Toward a Theoretical Explanation of Motivation Gains in Group Productivity, in E.H. Witte and J.H. Davis (Eds.) *Understanding Group Behavior: Consensual Action by Small Groups,* Erlbaum, Mahwah, NJ, 37-65.
27. Williams, K.D., Harkins, S.G., and Karau, S.J. (2003) Social Performance, in M.A. Hogg and J. Cooper (Eds.) *The SAGE Handbook of Social Psychology,* Sage Publications, Thousand Oaks, CA, 494-511.
28. Williams, K.D., and Karau, S.J. (1991) Social Loafing and Social Compensation: The Effects of Expectations of Co-Worker Performance, *Journal of Personality and Social Psychology*, 61, 4, 570-581.
29. Witte, E.H. (1989) Kohler Rediscovered: the Anti-Ringelmann Effect, *European Journal of Social Psychology*, 19, 2, 147-154.
30. Zajonc, R.B. (1980) Copresence, in P.B. Paulus (Ed.) *Psychology of Group Behavior,* Erlbaum, Hillsdale, NJ, 35-60.