Personality and the Nature of Collaboration in Pair Programming

Thorbjørn Walle and Jo E. Hannay
Simula Research Laboratory, Dept. of Soft. Eng., Pb. 134, 1325 Lysaker, Norway
Univ. of Oslo, Dept. of Informatics, Pb. 1080 Blindern, 0316 Oslo, Norway
thorbjw@ifi.uio.no, johannay@simula.no

Abstract

The benefits of synergistic collaboration are at the heart of arguments in favor of pair programming. However, empirical studies usually investigate direct effects of various factors on pair programming performance without looking into the details of collaboration. This paper reports from an empirical study that (1) investigated the nature of pair programming collaboration, and (2) subsequently investigated postulated effects of personality on pair programming collaboration. Audio recordings of 44 professional programmer pairs were categorized according to a taxonomy of collaboration. We then measured postulated relationships between the collaboration categories and the personality of the individuals in the pairs. We found evidence that personality generally affects the type of collaboration that occurs in pairs, and that different levels of a given personality trait between two pair members increases the amount of communication-intensive collaboration exhibited by a pair.

1. Introduction

Pair programming involves two programmers collaborating in front of one computer on the same programming task [6, 15, 47]. Several variants of pair programming are possible and practiced. For example, pairs may work for shorter or longer periods of time, partners may rotate, and driver and navigator roles may, or may not be adhered to. On the one hand, pair programming inspires a particularly close form of collaboration which might intensify group dynamics, while, on the other hand, short sessions may not allow more inert group dynamics to manifest themselves [17].

In any event, it is of interest to investigate factors that may affect the interaction that occurs in pair programming [1, 19]. These factors include personality, gender, expertise, attitudes, motivation, and preferences. However, since performance, e.g., in terms of time and quality, is often the ultimate criterion variable in software engineering, such factors have mostly been studied in terms of how much they directly influence performance, or in some cases, satisfaction [2, 12, 13, 16, 24, 26, 25, 31, 33, 41, 42, 44, 49]. This means that the nature of collaboration in terms of how pairs interact has mainly been treated in a black-box manner, with a few exceptions [8, 9, 10, 14, 18, 41].

When it comes to personality, the direct impact of personality on performance has been found to be modest in several areas of research [4, 38, 7], including software engineering [25]. However, even though direct effects on performance are disappointing, it is not unreasonable to expect that personality might have a more substantial impact on how pairs collaborate. How pairs collaborate might then be used to predict performance. Introducing pair collaboration as a mediator variable in this manner may then even reveal larger effects of personality on performance, since the effects of personality are filtered through the effects of pair collaboration, see Figure 1.

This paper focuses on the first part of this relationship, which consist of two issues: (1) the definition of the construct of pair collaboration, and (2) the relationship between personality and pair collaboration; for example, whether extroverts talk more, whether conscientious people have more task-focused conversation, and whether people with low emotional stability have more conflicts in collaboration. For (1), to avoid confounding of constructs, it is important to define the construct of pair collaboration before relating this construct to performance: Good and bad pair collaboration should not merely be defined to be whatever gives good and bad performance. If we were to do that, we would not gain insight into collaboration, and pair collaboration as a mediator variable would add nothing to the model. Therefore, the part of the relationship that concerns the effect of pair

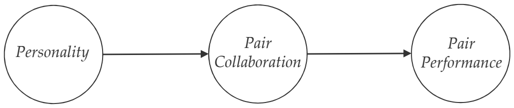

Figure 1. Pair Collaboration as a mediator variable

collaboration on pair performance is left for future research.

Audio recordings of 44 pairs solving a change task were analyzed. We developed a taxonomy for classifying pair collaboration from verbal interaction, that was intended to capture the nature of collaboration in terms of (a) which subtasks are performed, (b) what kind of interaction occurs, (c) on which cognitive level collaboration is conducted. We also recorded the extent of collaboration, that is, how much collaboration there is in pair programming. We then conducted an analysis to investigate postulated relationships that personality might have on collaboration.

Section 2 summarizes the study. Section 3 describes our investigation into pair collaboration, Section 4 describes the personality model that was used in this study, Section 5 presents the analysis of the effect of personality on pair collaboration, and Section 6 discusses and concludes.

2. Overview of Study

Our study used data collected during the experiment reported in [2], which compared the performance of professional pair programmers with that of solo programmers. Our study focuses on 44 of the pairs of that experiment. These programmers were recruited from software consultancy companies in Norway and Sweden in 2004/2005.

The pairs were formed so that both individuals in a pair had the same level of expertise. The subjects did not know in advance who their partner would be during the study. Within each level of expertise, pairs were assigned randomly to one of two treatments pertaining to task complexity. Note, however, that our present analysis does not include expertise or task complexity.

Each pair participated for one day and their session was divided into four stages. First, the subjects were given a presentation that included an introduction to the concept of pair programming, which focused on the active collaboration in pair programming and which involved a short description of the two roles (driver and navigator). The subjects were told that they could decide for themselves how often and when to switch roles, but that they had to try both roles at least once. After the presentation, the subjects started performing a training task, and a pretest task individually. Then, the subjects performed the three pair programming tasks T_1–T_3 as well as a time sink task T_4 in pairs. The pair programming tasks were done on two different versions of the system (a coffee machine application) according to the task complexity treatment. The first two tasks, T_1 and T_2, were simpler warm-up tasks (implementing a coin return button and adding a new drink to the menu). The third task T_3 was the main task (adding an ingredients check). To support the logistics of the study, the subjects used a web-based experiment support tool [3] to answer questionnaires, download code and documents, and to upload task solutions. For each task a test case was provided that each subject or pair used to test the solution. Further details, including validity issues, are provided in [2, 25]. All verbal interaction during tasks T_1–T_4 was audio recorded. At the end, the option was given to complete the Big Five personality test.

3. Collaboration

Collaboration denotes a situation in which both parties contribute new information to a given task. In contrast, *cooperation* involves splitting the task into subtasks and working on the subtasks separately [9]. Clearly, pair programming is intended as a collaborative task, rather than a cooperative task in this respect.

In order to determine the extent and nature of collaboration, we carried out a content analysis of the audio recordings. The objective of a content analysis is to elicit semantic content from recorded or written material in a systematic manner [32]. In our case, the semantic content of interest was the type of collaboration, and the analysis was done by categorizing passages of speech (the units of analysis) according to a coding scheme. We followed the steps of content analysis as summarized in [40]. The material consisted of the audio recordings of from the four pair programming tasks. We content analyzed the third task T_3 only. This decision was made upon the assumption that collaboration on this task was most representative for pair programming, since the pairs would have had a better chance to adjust to each other as well as to any problems with the experimental routines, equipment, etc. The pairs' team process may also have had some time to settle [36, 35] or "jell" [48].

3.1. Sampling Strategy

The first step of a content analysis is to determine the material to be analyzed. Our raw material was audio recordings. We faced a decision as to whether to transcribe the audio recordings to text. As discussed in [34], transcription is extremely time consuming and should be subject to cost-effectiveness considerations. In our case, the focus was on the content of collaborative measures, rather than on what was literally said. Exact transcripts were therefore regarded as less important, and we chose to code directly on the audio files using Transana as a coding tool.[1]

3.2. Units of analysis

Units of analysis were defined on two levels. The first level captured the task focus of the pair. A unit of analysis at this level spanned longer discourse sequences evidencing

[1] Transana is a trademark of the Wisconsin Center for Education Research, University of Wisconsin—Madison.

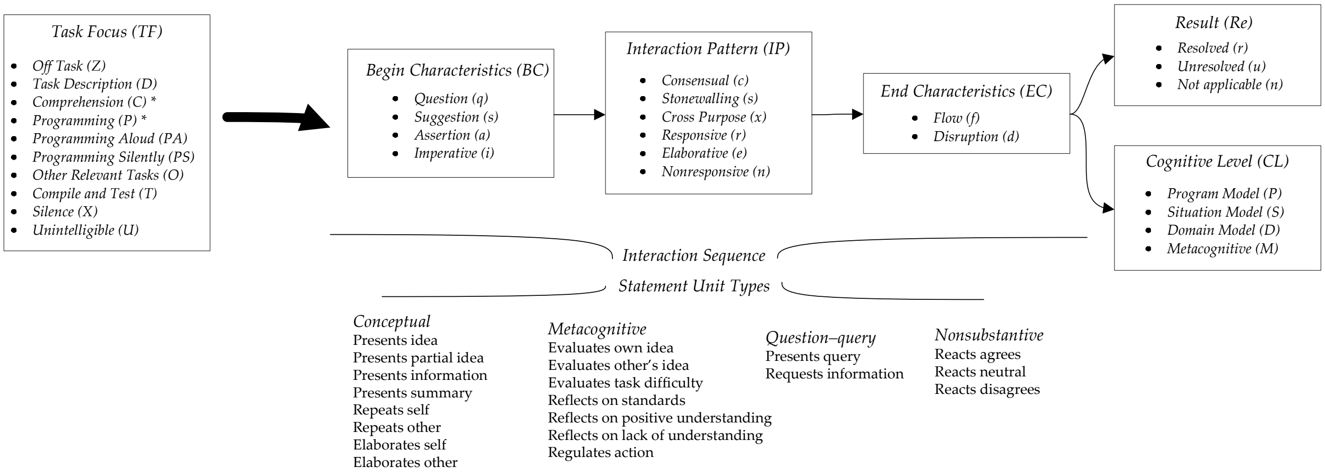

Figure 2. Coding schema

a particular focus until it was evident that a different focus was present (Section 3.3.1). Depending on the focus in question, a further analysis was conducted where units of analysis were so-called *interaction sequences* built from *statement units*. These interaction sequences were typically sequences of two or more alternating utterances between the peers on a certain topic (Section 3.3.2).

3.3. Themes

A content analysis revolves around a system of categories. Each subsystem in such a system is called a *theme*. Figure 2 shows our six themes. For example, the theme *Interaction Pattern* categorizes verbal discourse according to categories intended to describe collaboration. In this way, themes represent the constructs of the study. Constructs may correspond to concepts (of a theory), and in that case, the constructs (themes) may be predefined. Or constructs may emerge from a particular study, in which case themes are constructed in an exploratory manner in the manner of grounded theory building.

Our goal was to uncover patterns of collaboration, something we expected to be more universal than that which pertains to this particular study. In the outset, then, we wished to use predefined themes which capture the essence of collaboration in pair programming from verbal data. In our review of related work, we found several useful coding schemes, but we also realized that 1) no single scheme captured the aspects that we wanted to capture, 2) that several of the schemes did have several aspects but that they were not dealt with orthogonally, and 3) that coding schemes from other disciplines were also highly relevant.

We therefore used a hybrid approach, in a double sense: First, we combined coding schemes from several studies and from several disciplines. Second, we approached theme building in both a data-independent manner and in an exploratory manner in parallel: One preliminary coding scheme was developed on the basis of existing relevant coding schemes. Another preliminary coding scheme was developed on the basis of samples of our audio recordings. All coding schemes were subsequently systematized in a large table in order to show overlaps between categories and differences. This resulted in a distilled coding scheme, which was, as is customary [37], tested for reliability on 10 percent of the material. The resulting coding scheme is depicted in Figure 2, which we will now explain.

3.3.1 Task Focus

The categorization of *Task Focus* is intended to capture what the pair programmers are occupied with during a collaborative session. The starting point for defining this theme was a division of the task of pair programming into subtasks. Bryant *et al.* [9] divided pair programming into subtasks and registered the number of verbal utterances (in which new information was contributed) for each subtask. Our *Task Focus* is a coarser-grained and extended variant of Bryant *et al.*'s subtask categorization, and is shown to the left in Figure 2. The categories are defined as follows:

- *Off Task*—Utterances do not concern anything directly related to solving the task.
- *Task Description*—Utterances pertain to reading the task description and understanding what is to be done.
- *Code Comprehension*—Utterances pertain directly to solving the task with a focus on understanding existing code.
- *Programming*—Utterances pertain directly to solving the task with a focus on developing new code.
- *Programming Aloud*—Utterances made during active programming but not intended for dialog.

- *Programming Silently*—Active programming with no verbal utterances, i.e., only audible typing.
- *Other Relevant Tasks*—Utterances pertain to other tasks that are relevant to solving the main task, but that are not directly focused on code comprehension or programming.
- *Compile and Test*—Utterances pertain to compiling and testing the produced code.
- *Silence*—There are no utterances and no typing.
- *Unintelligible*—The utterances are not intelligible.

We are interested in problem-solving during actual programming. Thus, a further analysis was only conducted when *Task Focus* was *Code Comprehension* or *Programming* (although the other foci are also relevant). We analyzed the *amount* of actual verbal collaboration when programming in contrast to non-collaborative modes characterized by utterances not intended for collaboration or near-solo programming (hence the distinction between *Programming*, *Programming Aloud*, and *Programming Silently*).

3.3.2 Interaction Sequences

If the *Task Focus* was *Code Comprehension* or *Programming*, a further analysis was conducted in order to determine *interaction sequences* (Figure 2 middle part). The basic units of an interaction sequence are *statement units*. A statement unit is a "codable unit of speech (i.e., a word, sentence, or sentences)" [27]. An interaction sequence is then built from one or more statement units.

Hogan *et al.* [27] assigns each statement unit to a *Statement Unit Type* under main categories *Conceptual*, *Metacognitive*, *Question-query*, and *Nonsubstantive* (Figure 2 lower part). Statement units of certain types typically begin interaction sequences (e.g., Presents idea, Requests information), while others typically end interaction sequences (e.g., Repeats other, Reacts agrees). Some types may both begin and end an interaction sequence, and statements of other types will typically only be found in the middle of interaction sequences (e.g., Elaborate other).

An interaction sequence begins when *Task Focus* switches to *Comprehension* or *Programming*, or when a preceding interaction sequence ends. An interaction sequence ends when it is immediately replaced by another interaction sequence, or when a *Task Focus* category other than *Comprehension* or *Programming* starts. The difficult part of this is to determine when one interaction sequence ends and another begins. The main guideline for determining the transition from one interaction sequence to the next is a change of topic. However, any change in the nature of discourse also warrants the end of one sequence and the beginning of another, that is, if it is reasonable to define a piece of discourse into two interaction sequences with two different characteristics, then this warrants the existence of these two interaction sequences instead of a single sequence.

Interaction sequences are characterized according to the five themes *Begin Characteristics*, *Interaction Pattern*, *End Characteristics*, *Result*, and *Cognitive Level*. The themes are coarser grained and at a higher level of abstraction than the main categories of the Statement Unit Types. Nevertheless, statement units types are aids to indicate which theme, and which category in a theme, that is appropriate.

3.3.3 Begin Characteristics

This theme characterizes the first statement unit of an interaction sequence, and is designed to capture degrees of assertiveness.

- *Question*—A passive request for information/clarification.
- *Suggestion*—An active contribution of an idea, a plan of action, or information; possibly formulated as a question, e.g., a statement unit type of Presents idea or similar.
- *Assertion*—A statement describing or claiming how things are or how things should be done), e.g., a statement unit type of Presents idea, Presents Information, or similar.
- *Imperative*—A statement asking or ordering the other peer to do something.

3.3.4 Interaction Patterns

This theme characterizes the interaction sequence as a whole, and is designed to capture the central aspects of collaboration. Our categories are modeled around Hogan *et al.*'s [27] interaction pattern categories. The categories are defined as follows:

- *Consensual*—Only one speaker contributes substantive statements (i.e., *Conceptual*, *Metacognitive*, *Question-queries*), while the other speaker responds by (a) agreeing with the statement, passively or neutrally acknowledging the statement, (b) actively accepting what was said and thereby encouraging the speaker to continue, or (c) repeating the preceding statement verbatim. Thus, in consensual sequences one speaker carries the conversation, with the other speaker serving as a minimally verbally active audience. Although consensual sequences may last only a few statement units, sometimes a single speaker may contribute many ideas to the discussion with all of the intervening statements by the other speaker being nonsubstantive.
- *Stonewalling*—A speaker delays a response or otherwise obstructs any elaboration of the other peer's input.
- *Cross Purpose*— The speakers' statements are cross purpose, that is, each speaker is speaking on separate topics.
- *Responsive*—Both questions and responses of both peers contribute substantive statements to the discussion. Responsive patterns are often only a few statement units in length. They become longer when several agreements or neutral comments are embedded within the sequence.

- *Elaborative*—Both peers contribute substantive statements to the discussion, as in the responsive sequences. Moreover, the speakers make multiple contributions that build on or clarify another's prior statement. Elaborations are coconstructive additions (linking a new idea to someone else's idea or partial idea), corrections (correcting someone's statement with a simple, undisputed statement), or dialectical exchanges (disagreeing with the prior statement and offering a counterargument).

The difference between *Responsive* and *Elaborative* lies in the greater length and the joint collaborative building of understanding of the *Elaborative* interaction pattern. *Elaborative* interaction sequences may also tend to be more balanced with regards to dominance between the peers.

- *Nonresponsive*—A statement is met with no response.

The categories *Stonewalling*, *Cross Purpose* and *Nonresponsive* are not parts of Hogan *et al.*'s [27] schema. *Stonewalling* is a category that arose from our data, but it's name is taken from Chan [11], where this category characterizes a rejection of differences to minimize belief change. *Nonresponsive* also arose from our data.

In all of the above categories, we also included nonverbal actions where appropriate and unambiguous. For example, an acknowledging "*mm-mm*" would mean that the interaction sequence would be classified as *Consensual*.

3.3.5 End Characteristics

This theme characterizes the end of an interaction sequence, and is designed to capture whether collaboration flows naturally or is disrupted [29, 30]. It is not linked as specifically to the statement units of an interaction sequence as are *Begin Characteristics*. *End Characteristics*, such as *Disruption*, may also depend on earlier statements in the sequence or on the beginning of the next interaction sequence.

- *Flow*—An interaction sequence ends by flow if the interaction ends naturally with no disruption.

- *Disruption*—An interaction sequence ends by disruption if a speaker changes the topic.

3.3.6 Result

An interaction sequence may lead to an issue becoming resolved or it may lead to no clear conclusion.

- *Resolved*—An interaction sequence is resolved if it leads to a plan of action, or if the initiating question, suggestion, or assertion is answered or otherwise resolved.

- *Unresolved*—An interaction sequence is unresolved if there is no clear plan of action and the initiating question, suggestion, or assertion is not answered or resolved.

- *Not applicable*—if neither of the above are applicable.

3.3.7 Cognitive Level

The cognitive level of verbal utterances reflects problem-solving strategies. The basis for our *Cognitive Level* theme is the first level of von Mayrhauser and Lang's AFECS coding scheme [45]. AFECS is a step toward standardizing protocol analysis for observing programming behavior. It incorporates the mental models framework for program comprehension developed by [46]. In object-oriented programming, objects are central entities that map to the problem domain, hence providing the programmer with a link between program model and domain model.

- *Program Model*—Statements pertaining to elementary operations, elementary functions, or control flow.

- *Situation Model*—Statements pertaining to the classes and objects and the relations between them. Statements pertaining to main goals of the programming tasks.

- *Domain Model*—Statements pertaining to the real-world domain of the program development effort.

- *Metacognitive*—Statements pertaining to the actual solving of the task, that is, assessments pertaining to progress, understanding, collaboration etc.

3.4. Final Analysis

Four coders, including the first author, classified the audio recordings in Transana according to the coding scheme in Section 3.3. The coders had no prior experience with this type of analysis. They went through a period of practice with the tools and methods involved prior to the coding process. Two of the coders were associated with the research, while the other two were hired to do coding exclusively. Sections of the recordings (clips) were allocated to categories complete with time stamps. The coders worked in pairs. One coder was assigned the responsibility for each audio file, and first determined the beginning and end points of T_3 in the audio recording. Then, both coders coded five minutes near the beginning of T_3 independently, and subsequently discussed differences in their codings until they agreed on a joint coding. Finally, the coder responsible for the file coded the rest of T_3 alone, and subsequently submitted the completed coding to the other coder for a verification check of a random five minutes toward the end of T_3. Differences were discussed until agreement was reached. The initial joint coding of the first five-minute section was found to be a necessary calibration procedure to ensure that the main coder would have some frame of reference for the rest of the coding session. This was particularly useful for getting accustomed to verbal style. An aggregated inter-rater agreement score was calculated based on the theme structure. The score was computed on the independent codings (prior to discussion) of the two five-minute portions of the audio file that were analyzed by both coders. The agreement scores were in the range of 78%–92%, which is acceptable.

4. Personality

There exist several models of personality with several alternative operationalizations or tests (usually questionnaires) that are administered to measure a person's personality. A model that in recent years has dominated the academic scene [4] consist of five factors and goes under the name of the *Big Five* [20, 21]. The Big Five posits that the most important personality differences in people's lives will become encoded as terms in their natural language, the so-called *Lexical Hypothesis* [20].

The Big Five model consists of the following five personality factors (traits) [20, 21] (with descriptions from [39]):

Extraversion (Factor 1) Assesses quantity and intensity of interpersonal interaction; activity level; need for stimulation; and capacity for joy.

Agreeableness (Factor 2) Assesses the quality of interpersonal orientation along a continuum from compassion to antagonism in thoughts, feelings, and actions.

Conscientiousness (Factor 3) Assesses degree of organization, persistence, and motivation in goal-directed behavior. Contrasts dependable, fastidious people with those who are lackadaisical and sloppy.

Emotional stability/Neuroticism (Factor 4) Assesses adjustment versus emotional stability. Identifies proneness to psychological distress, unrealistic ideas, excessive urges, and maladaptive coping responses.

Openness to experience (Factor 5) Assesses proactive seeking and appreciation of experience for its own sake; toleration for and exploration of the unfamiliar.

We used the Big Five Factor Markers [28, 23, 22] with 100 indicators—20 per trait. The 100 indicators are self-assessment questionnaire items on a seven-point Likert scale. For example, one of the 20 items for *Extraversion* is "I feel comfortable around people", and one of the 20 items for *Agreeableness* is "I make people feel at ease".

For pairs, some notion of *Pair Personality* must be devised. The most common ways of aggregating personality scores into team scores is by taking the mean, the minimum, the maximum, or the variance of the individual scores. These aggregates can be seen as alternative operationalizations of two *Pair Personality* constructs, namely trait *Elevation* (with the mean as the canonical measure) and *Variability* (with the variance, or difference as the canonical measure) [5, 25, 38, 43].

5. Personality and Collaboration

We postulated the following relationships:

R_1 *Personality* affects the type of *Collaboration*. This relationship investigates our overall exploratory research question, and concerns the initial usefulness of *Collaboration* as a mediator variable as depicted in Figure 1.

R_2 *Variability* in *Personality* increases the amount of communication-intensive *Collaboration*. This is the postulate most closely based on findings in the available literature. It is primarily based on findings from the Myers-Briggs Type Indicator-based pair programming and collaboration experiment by Sfetsos *et al.* [41]. The study indicated that there is a significant correlation between mixed personalities and high amounts of communication transactions. Their results indicate that pairs with mixed personalities both communicate better and perform better. Further, Karn and Cowling claim that homogeneous teams (with respect to personality) are not ideal, and that such teams "run into a real danger of falling into the no debate trap" [29]. Williams *et al.* [49] present a similar finding.

R_3 Pairs in which the members have similar levels of *Extraversion* are less likely to disrupt each other. This relationship surfaced in the ethnographic study in [29]. In one of the groups in the study, none of the members disrupted anyone else at all. The authors explain this by the fact that the group had one clearly dominant member, the group's only extrovert.

R_4 High *Extraversion Elevation* leads to more communication-intensive *Collaboration*. This relationship is suggested in [47]. Two extraverts will talk unnecessarily, sometimes about things outside the task, and will thus spend longer time on the task.

R_5 High *Agreeableness Elevation* leads to more *Off Task* communication. This is an assumption based on the definitions of the personality traits and our collaboration categories. It is reasonable to believe that people scoring high on agreeableness might initiate off task communication such as small talk, since agreeableness indicates a genuine interest in other people's lives.

R_6 High *Extraversion Elevation* leads to more *Metacognitive* statements. This is, like R_5 based on the definitions of the personality traits and our collaboration categories. Extraverts might be more likely to cope with frustration about tasks by expressing their opinions on and feelings about the task.

5.1. Analysis

The next sections describe the analysis of these relationships. Due to probable nonlinear effects of personality [25], we used decision tree analysis as implemented in *jmp* 7. Data files were prepared with SAS 9.2, Enterprise Guide 4 and SPSS 16.2 as in [25], together with additional scripts.[2]

5.1.1 Variables

The starting point of a decision tree analysis is a dependent variable and a set of independent variables. The independent variables in our case were operationalizations of *Pair Personality Elevation* and *Variability* (Section 4) in terms of score means and differences. The dependent variables were measures of the various theme categories for *Collaboration* (Section 3.3). We used two types of measure: The first type was the percentage of time allocated to a category, relative to the total length of T_3; for example, the amount of time spent by a pair making utterances that were classified to, say, *Task Description* under *Task Focus* (*TF-D*). The second type of measure was the percentage of clips allocated to a category. For interaction sequence categories, the percentage was calculated relative to the total number of interaction sequence clips for a pair; for example, the percentage of *Elaborative* interaction patterns relative to the total number of clips classified as interaction sequences for a pair. For other clips, the percentage was calculated relative to the total number of clips for a pair.

5.1.2 Decision Trees

Decision tree analysis is an iterative process that successively splits the original *n* observations in a dependent variable in halves, thus creating a binary tree structure. Figure 3 shows the resulting decision tree for *Programming Aloud* (percentage of time) as the dependent variable. Here, these two splits involve the independent variables *Emotional Stability difference* (B5_4_StdDev) and *Extraversion mean* (B5_1_Mean). Any split is associated to an independent variable such that all observations in one partition are less in the independent variable than all observations in the other partition (for ordinal or continuous independent variable), or according to categories (for categorical independent variables). Each split is chosen as the one that maximizes some split criterion. In our case, that criterion is to maximize the significance of the resulting difference in the dependent variable. In order to obtain a reasonable robustness toward outliers, the process was set to terminate when partition sizes went below 5.

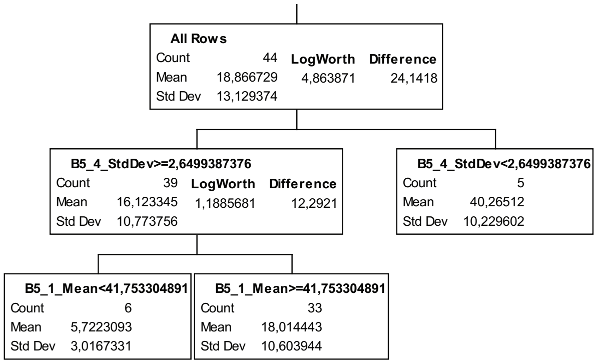

Figure 3. Example of decision tree

Decision tree analysis is independent of any assumptions on normality or types of data. Splits nearer the root split more of the observations and signify more general effects than splits further away from the root. Nonlinear effects are reflected by successive and asymmetrical splits (in the sense of producing an unbalanced tree) with respect to the same independent variable. Interaction effects are reflected by asymmetrical splits of different variables.

In Figure 3, only the top split is a significant split.[3] The tree should be read as follows (if focusing on all splits, regardless of significance): The main effect for the percentage of time used on *Programming Aloud* was in the *difference* of *Emotional Stability* (B5_4_StdDev), where less difference leads to more *Programming Aloud*. For the pairs who had higher differences in *Emotional Stability*, high scores on *Extraversion* (B5_1_Mean) leads to more *Programming Aloud*.

5.1.3 Analysis Procedure

A standard explorative mode of employing decision tree analysis is to split the data until the minimum split size is reached and to subsequently report the findings. We employed a slightly different mode. When a significant split appeared in the first/topmost split, the independent variable that the split was based on was removed and the analysis run again. This was done to allow potential significant first splits in other variables to surface. This procedure was repeated until no significant first splits appeared. Thus, for independent variables where the first split (when all explanatory variables were included) was not significant, no additional analysis runs were needed for that variable.

[2] SAS, Enterprise Guide, and *jmp* are trademarks of SAS Institute Inc. SPSS is a trademarks of SPSS Inc.

[3] The *LogWorth* index in parent node is a significance measure of the difference in mean values the child nodes with regards to the dependent variable. Specifically, LogWorth = $-log_{10}(p)$, where p is the adjusted probability of the observed data under the hypothesis of the means being equal. Thus, p-values less than 0.05 are reflected by LogWorth-values greater than 1.30.

	Support	Findings
R_1	Yes	Several personality factors influence collaboration categories significantly
R_2	Yes	Personality differences influence several communication-intensive collaboration categories significantly
R_3	No	Contradicting evidence
R_4	No	No significant impact found
R_5	No	Extraversion affects several communication-intensive collaboration categories, and mostly in one direction, but only non-significantly
R_6	No	No significant impact found

Table 1. Support for postulated relationships

This focus on top splits describes the most general trends, but in the presence of all independent variables. This procedure will not prevent potential significant top splits from surfacing, since the splits are made on the basis of maximizing significance. This means that the top-most split will always be more significant than the top-most split of the next run, where the variable that was in the previous top split is removed. When the top split is no longer significant, the process can be stopped, since the next top split (after removing yet a variable) would be *even less* significant. The process does, however, discard significant splits beneath non-significant top splits. This exclusion is justifiable, since significance is a criterion for the model.

In addition to this, the standard complete split trees were investigated with regards to the specific variables that are involved in the relationships. Often, this extra check lead to the realization that certain findings had to be revised.

5.2. Results

The relationships R_1–R_6 were operationalized in terms of the corresponding measures described in Section 5.1.1, and subjected to data analysis. The data analysis only supported R_1 and R_2, see Table 1. In the following we give a more detailed account of the findings.

R_1—*Personality* affects the type of *Collaboration*. Table 2 lists all the significant findings when using the procedure described in Section 5.1.3. The "+/-" columns indicate whether the collaboration category is affected positively or negatively by the personality factor. For example, the first row shows a relationship between *Extraversion mean* and the *Other Relevant Tasks* (*O*) category under the *Task Focus* theme (*TF*) (Section 3.3.1). The "-" in the third column indicates a negative relationship, i.e., that a high *Extraversion mean* score for the pair relates to less *TF-O*. Nearly all personality factors influenced one or more of the *Collaboration* category measures significantly, both with respect to percentage of time allocated to a category, (Table 2 left part), and when measuring percentage of times a category was used (Table 2 right part). Thus, there is evidence in support of R_1.

Personality Measure	Time Allocated to Category			Times Category Used		
	% time	+/-	LogWorth (*p*)	% clips	+/-	LogWorth (*p*)
Extraversion mean	*TF-O*	-	1.43 (0.04)	*EC-f*	-	1.48 (0.03)
	CL-P	+	1.31 (0.05)	*EC-d*	+	1.48 (0.03)
				IP-n	-	1.37 (0.04)
Extraversion diff	*TF-Z*	+	3.94 (0.00)	*IP-x*	+	2.00 (0.01)
	IP-x	+	1.51 (0.03)			
Agreeableness mean	*IP-s*	+	2.48 (0.00)	*TF-P*	-	2.07 (0.01)
	BC-q	-	1.87 (0.01)	*TF-C*	+	2.07 (0.01)
	IP-x	+	1.65 (0.02)	*Re-r*	-	2.00 (0.01)
	Re-r	-	1.50 (0.03)	*IP-q*	-	1.97 (0.01)
	CL-D	-	1.47 (0.03)	*Re-u*	+	1.88 (0.01)
				CL-M	-	1.35 (0.04)
Agreeableness diff	*TF-X*	+	1.35 (0.04)	*Re-r*	-	1.47 (0.03)
Conscientiousness mean	*TF-D*	-	2.16 (0.01)			
Conscientiousness diff						
Emotional Stability mean	*TF-Z*	-	1.39 (0.04)			
	CL-M	+	1.59 (0.03)			
Emotional Stability diff	*TF-PA*	-	4.86 (0.00)			
Openness mean	*BC-i*	+	3.70 (0.00)	*BC-i*	+	4.68 (0.00)
Openness diff	*IP-r*	+	2.11 (0.01)	*IP-r*	+	1.60 (0.03)

Table 2. Significant *Personality* influences on *Collaboration* categories—top splits

R_2—*Variability* in *Personality* increases the amount of communication-intensive *Collaboration*. Communication-intensive *Collaboration* relates to the categories *Off Task* (*TF-Z*) *Elaborative* (*IP-e*), *Responsive* (*IP-r*), *Cross Purpose* (*IP-x*), *Disruption* (*EC-d*), and *Unresolved* (*Re-u*). These categories either signify substantial mutual verbal involvement from both parties (e.g., *Elaborative*) or an overflow of verbal initiative (e.g., *Disruption*).

Several of these collaboration categories were found to be significantly influenced by *difference* (the operationalization of *Variability*) in *Personality*. This supports R_2. Some categories, however, contradicted this, but mostly non-significantly. The only significant finding that contradicted R_2, can be seen in the left part of Table 2: For pairs with a high *Agreeableness difference*, the programmers are silent (*TF-X*) for a longer total time than pairs with more similar *Agreeableness* scores.

To investigate R_2 further, we extended the top split analysis described in Section 5.1.3 with a larger aggregated analysis similar to that in [25]. For this, the complete split trees were investigated, and the order of the splits, as well as whether they were significant or not, was noted. In addition to the above-mentioned communication-intensive categories, we designated a group of silent categories as a contrast. This group consisted of *Programming Silently* (*PS*), *Silence* (*TF-X*), *Nonresponsive* (*IP-n*), *Consensual* (*IP-c*), *Stonewalling* (*IP-s*), *Flow* (*EC-f*), and *Resolved* (*Re-r*).

We then performed the complete tree analyses on these two contrasting groups of categories. This analysis can be seen in Table 3. The numbers indicate how early the split occurred (with 1 being the first split of the tree, 2 the second, and so on). A '*' indicates a significant split. The +/- in front of the numbers indicate whether the split signifies a positive or a negative relationship. At the bottom of each column, the R^2 of an n-fold cross validation is given, that is, the average R^2 over n=44 predictions where $n-1$ observations are used to predict the nth observation. At the

	Communication-intensive Categories (C)											Silent Categories (S)											
	TF-Z	*IP-e*		*IP-r*		*Re-u*		*IP-x*		*EC-d*		*TF-PS*	*TF-X*	*IP-n*		*IP-c*		*IP-s*		*EC-f*		*Re-r*	
	Time	Time	Clips	Time	Clips	Time	Clips	Time	Clips	Time	Clips	Time	Time	Time	Clips	Time	Clips	Time	Clips	Time	Clips	Time	Clips
Extraversion diff	+1* -5 +6	+4*			+5	+1 -2 +4	+1 -2 +4	+1* -4	+1*	-6	-3 +4				-1*	-3 +6	+5	+2* +3	+2 +4	+6	+3 -4	+4	+5
Agreeableness diff		-1	-3 -6	+6		+5	+5	+2	+2		+5	+2	+1*	+4 -5	+2* -3 +4			+1 -4	+1 -3		-5	-1* +3	-1* +3
Conscientiousness diff	+3* +4	-5	-5	+5	+6	-3	-3					+1 -4 +5	+2 +5		-6 -7	-2	-1 -2 -6					+5	
Emotional Stability diff	+2* -7	-3*	-2*	+2	+1*				+3 -4	-2 -3 +4 -5*	-2	+3	-3 +4 +6	+1 -2*		-1 +4 +5				+2 +3 -4 +5*	+2		
Openness diff		-2	-1 +4	+1* +3 -4	+2 -3 +4			+3		+1	+1 +6	+6		+3 -6	+5		+3 -4			-1	-1 -6	-2	-2 +4
n -Fold R^2	.36	.46	.16	.32	.24	.12	.16	-.04	.04	-.04	-.04	.28	.32	.12	.40	.16	.20	-.09	-.14	-.04	-.04	.21	.21
Overall R^2	.56	.59	.37	.50	.48	.33	.36	.21	.25	.26	.30	.49	.54	.36	.59	.40	.44	.24	.21	.26	.30	.40	.40

Table 3. Exploratory analysis for R_2

very bottom, the overall R^2, which indicates the ratio of explained variance, is given.

Table 4 ranks each independent variable according to how early associated splits occurred, by the formula *max splits* + 1 - *split number*. In our case, *max splits* was seven, that is, no trees had more than seven splits. Thus, if, say, *Agreeableness diff* assumed a significant second split in a model, then that split would contribute 7+1-2=6 to the ranking for *Agreeableness diff*. The '*'-columns only count significant splits. The first two columns are the results from the communication-intensive categories (C) minus the results from the silent categories (S). This difference indicates the overall influence that the personality trait has on the amount of communication-intensive *Collaboration*. A positive score indicates that difference in personality traits leads to an increase in communication-intensive *Collaboration* and would give evidence in favor of R_2. A negative score would contradict R_2.

Table 3 and 4 show that there are indeed a number of splits that indicate that variability in personality has an impact on the communication-intensive categories. This is most evident for *Extraversion*, and it confirms our findings from the first analysis: The amount of communication-intensive *Collaboration* does increase when the pairs have different personalities (except for differences in emotional stability). This provides support in favor of R_2.

R_3—Pairs in which the members have similar levels of *Extraversion* are less likely to disrupt each other. Our analysis does not offer support for R_3. Individuals that have similar levels of *Extraversion* do not disrupt each other more, but there is no significant evidence that they will disrupt each other less either. The partition trees for this analysis are contradictory with signs of both positive and negative relationships. However, the top split analysis done in connection with R_1 does suggest (non-significantly) a positive relationship between *Extraversion difference* and *Disruption*. Further investigation of this relationship should be conducted, for example with an expanded sample size.

	(C-S)*	C-S	C*	C	S*	S
Extraversion diff	26	6	25	30	-1	24
Agreeableness diff	1	-5	0	12	-1	17
Conscientiousness diff	5	4	5	-2	0	-6
Emotional Stability diff	2	-31	-1	-8	-3	23
Openness diff	7	40	7	25	0	-15

Table 4. Exploratory analysis for R_2 aggregated

R_4—High *Extraversion Elevation* leads to more communication-intensive *Collaboration*. Our analysis does not offer support for R_4. Analyses seem to suggest that communication-intensive *Collaboration* categories are affected by *Extraversion*, and mostly in one direction, but the results are highly non-significant on all categories. The only significant finding was that extraverts will disrupt each other more often than introverts.

R_5—High *Agreeableness Elevation* leads to more *Off Task* communication. Our analysis does not offer support for R_5. *Agreeableness mean* does not significantly influence time used on neither *Off-Task*, *Metacognitive* nor *Other Relevant Tasks*. (The non-significant results suggest that *Agreeableness mean* decreases occurrences in all three categories.) Moreover, high *Agreeableness mean* significantly relates to fewer occurrences of *Metacognitive* statements. So if anything, highly agreeable people small talk less than their not-so-agreeable counterparts in a pair programming situation. This suggests that agreeable people will use the cognitive level of *Metacognitive* less.

R_6—High *Extraversion Elevation* leads to more *Metacognitive* statements. Our analysis does not offer support for R_6. The significance of the splits are very low, and the split trees are contradictory. Our analysis gives no reason to posit that *Extraversion* has any effect on the amount of use of the cognitive level *Metacognitive*.

5.3. Threats to Validity

The two most important threats to validity of this study are construct validity and the corresponding (inter-rater) measurement reliability. The structure of the collaboration construct should be investigated further in terms of existing and additional sub-constructs (themes), and the corresponding operationalizations should be validated and verified on larger samples. The omission of moderator variables such as *Task Complexity* and *Expertise* is a threat to internal validity. Statistical conclusion validity is threatened by the exploratory nature of our study, and further analyses should be conducted in a confirmatory manner using complementary methods. In order to increase the study's external validity, replications on variants of this study's variables should be conducted. Validity issues regarding the personality test and the experimental setting are summarized in [2, 25].

6. Discussion and Conclusion

This study investigated postulated relationships between personality and the nature of pair collaboration. The latter was coded according to a scheme of thematic collaboration categories. Although several of the more specific relationships were not supported by our data analysis, we found that almost all of the personality factors lead to a significant increase or decrease with regards to at least one collaboration category. We also found that pairs consisting of two people with different levels on a personality trait will communicate more. However, we also found that differences in certain personality factors affected collaboration differently than others, and that a large difference in every single factor of personality is not necessarily beneficial to collaboration.

These initial results thus suggest that personality might affect pair collaboration, and that the impact of personality on pair collaboration may be more visible than the impact on pair performance.

Our collaboration construct and its operationalization were derived from existing theoretical and empirical results, and our study showed that the construct was capable of discerning various types of collaboration. The next step in this line of research is to refine the construct of collaboration and its operationalization. As mentioned at the beginning of this paper, once the collaboration construct has gone through refinement, its role as a mediator of personality to performance can be investigated. In this wider context, well-known moderator variables such as expertise and task complexity should be included in the model as well.

Acknowledgments

The authors are grateful to Bente Nordbø, Stian Friberg, Alexander Ottesen, and Martha Kutscha for assistance in analyzing audio files and to Vigdis By Kampenes and Hans Christian Benestad for invaluable discussions.

References

[1] M. Ally, F. Darroch, and M. Toleman. A framework for understanding the factors influencing pair programming success. In *Proc. XP 2005*, pages 82–91. Springer-Verlag, 2005.

[2] E. Arisholm, H. Gallis, T. Dybå, and D.I.K. Sjøberg. Evaluating pair programming with respect to system complexity and programmer expertise. *IEEE Trans. Software Eng.*, 33:65–86, Feb. 2007.

[3] E. Arisholm, D.I.K. Sjøberg, G.J. Carelius, and Y. Lindsjørn. A web-based support environment for software engineering experiments. *Nordic J. Computing*, 9(4):231–247, 2002.

[4] M.B. Barrick, M.K. Mount, and T.A. Judge. Personality and performance at the beginning of the new millennium: What do we know and where do we go next? *Int'l J. Selection and Assessment*, 9(1/2):9–30, 2001.

[5] M.R. Barrick, G.L. Stewart, M.J. Neubert, and M.K. Mount. Relating member ability and personality to work-team processes and team effectiveness. *J. Applied Psychology*, 83(3):377–391, 1998.

[6] K. Beck and C. Andres. *Extreme Programming Explained: Embrace Change*. Addison-Wesley, second edition, 2003.

[7] S.T. Bell. Deep-level composition variables as predictors of team performance: A meta-analysis. *J. Applied Psychology*, 92(3):595–615, 2007.

[8] S. Bryant. Double trouble: Mixing qualitative and quantitative methods in the study of eXtreme Programmers. In *Proc. 2004 IEEE Symp. on Visual Languages and Human Centric Computing*. IEEE Computer Society, 2004.

[9] S. Bryant, P. Romero, and B. du Boulay. The collaborative nature of pair programming. In *Proc. XP 2006*, pages 53–64. Springer-Verlag, 2006.

[10] L. Cao and P. Xu. Activity patterns of pair programming. In *Proc. 38th Annual Hawaii Int'l Conf. System Sciences*, pages 1–10. IEEE Computer Society, 2005.

[11] C.K.K. Chan. Peer collaboration and discourse patterns in learning from incompatible information. *Instructional Science*, 29:443–479, 2001.

[12] J. Chao and G. Atli. Critical personality traits in successful pair programming. In *Proc. AGILE 2006*. IEEE Computer Society, 2006.

[13] K.S. Choi. *A Discovery and Analysis of Influencing Factors of Pair Programming*. PhD thesis, Faculty of New Jersey Institute of Technology, Department of Information Systems, 2004.

[14] J. Chong and T. Hurlbutt. The social dynamics of pair programming. In *Proc. 29th Int'l Conf. Software Engineering*, 2007.

[15] L.L. Constantine. *Constantine on Peopleware*. Prentice Hall, 1995.

[16] A.J. Dick and B. Zarnett. Paired programming & personality traits. In *Proc. Third Int'l Conf. Extreme Programming and Agile Processes in Software Engineering (XP 2002)*, pages 82–85, 2002.

[17] D.R. Forsyth. *Group Dynamics*. Thomson Wadsworth, fourth edition, 2006.

[18] S. Freudenberg (née Bryant), P. Romero, and B. du Boulay. 'talking the talk': Is intermediate-level conversation the key to the pair programming success story? In *Proc. AGILE 2007*. IEEE Computer Society, 2007.

[19] H. Gallis, E. Arisholm, and T. Dybå. An initial framework for research on pair programming. In *Proc. 2003 Int'l Symp. Empirical Software Engineering (ISESE'03)*, pages 132–142, 2003.

[20] L.R. Goldberg. An alternative description of personality: The big-five factor structure. *J. Personality and Social Psychology*, 59:1216–1229, 1990.

[21] L.R. Goldberg. The structure of phenotypic personality traits. *American Psychologist*, 48:26–34, 1993.

[22] L.R. Goldberg. A broad-bandwidth, public domain, personality inventory measuring the lower-level facets of several five-factor models. In I. Mervielde, I. Deary, F.D. Fruyt, and F. Ostendorf, editors, *Personality Psychology in Europe*, volume 7, pages 7–28. Tilburg University Press, 1999.

[23] L.R. Goldberg, J.A. Johnson, H.W. Eber, R. Hogan, M.C. Ashton, C.R. Cloninger, and H.C. Gough. The international personality item pool and the future of public-domain personality measures. *J. Research in Personality*, 40:84–96, 2006.

[24] B. Hanks. Student attitudes toward pair programming. In *Proc. 11th Annual Conf. Innovation and Technology in Computer Science Education (ITiCSE06)*, pages 113–117. ACM, 2006.

[25] J.E. Hannay, E. Arisholm, H. Engvik, and D.I.K. Sjøberg. Personality and pair programming. *To appear in IEEE Transactions on Software Engineering*, 2009.

[26] J.E. Hannay, T. Dybå, E. Arisholm, and D.I.K. Sjøberg. The effectiveness of pair programming: A meta-analysis. *To appear in Information & Software Technology*, 2009.

[27] K. Hogan, B.K. Nastasi, and M. Pressley. Discourse patterns and collaborative scientific reasoning in peer and teacher-guided discussions. *Cognition and Instruction*, 17(4):379–432, 2000.

[28] International Personality Item Pool. A scientific collaboratory for the development of advanced measures of personality traits and other individual differences, 2007.

[29] J.S. Karn and A.J. Cowling. A study of the effect of personality on the performance of software engineering teams. In *Proc. Fourth Int'l Symp. Empirical Software Engineering (ISESE'05)*, pages 417–427. ACM, 2005.

[30] J.S. Karn and A.J. Cowling. A follow up study of the effect of personality on the performance of software engineering teams. In *Proc. Fifth Int'l Symp. Empirical Software Engineering (ISESE'06)*, pages 232–241. ACM, 2006.

[31] N. Katira, L. Williams, E. Wiebe, C. Miller, S. Balik, and E. Gehringer. On understanding compatibility of student pair programmers. In *Proc. 35th Technical Symp. Computer Science Education (SIGCSE'04)*, pages 7–11. ACM, 2004.

[32] K. Krippendorff. *Content Analysis: An Introduction to its Methodology*. Sage, second edition, 2004.

[33] L. Layman. Changing students' perceptions: An analysis of the supplementary benefits of collaborative software development. In *Proc. 19th Conf. Software Engineering Education and Training (CSEET'06)*. IEEE Computer Society, 2006.

[34] T. Lethbridge, S.E. Sim, and J. Singer. Studying software engineers: Data collection techniques for software field studies. *Empirical Software Engineering*, 10:311–341, 2005.

[35] L.L. Levesque, J.M. Wilson, and D.R. Wholey. Cognitive divergence and shared mental models in software development project teams. *J. Organizational Behavior*, 22:135–144, 2001.

[36] J.E. Mathieu, G.F. Goodwin, T.S. Heffner, E. Salas, and J.A. Cannon-Bowers. The influence of shared mental models on team process and performance. *J. Applied Psychology*, 85(2):273–283, 2000.

[37] P. Mayring. Qualitative content analysis. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research [On-line Journal]*, 1(2), June 2000. Available at: `http://www.qualitative-research.net/fqs-texte/2-00/2-00mayring-e.htm`.

[38] M.A.G. Peeters, H.F.J.M. van Tuijl, C.G. Rutte, and I.M.M.J. Reymen. Personality and team performance: A meta-analysis. *European J. of Personality*, 20:377–396, 2006.

[39] L.A. Pervin and O.P. John. *Personality: Theory and Research*. John Wiley & Sons, Inc., seventh edition, 1997.

[40] C. Robson. *Real World Research*. Blackwell Publishing, second edition, 2002.

[41] P. Sfetsos, I. Stamelos, L. Angelis, and I. Deligiannis. Investigating the impact of personality types on communication and collaboration-viability in pair programming—an empirical study. In *Proc. Seventh Int'l Conf. Extreme Programming and Agile Processes in Software Engineering (XP 2006)*, volume 4044 of *Lecture Notes in Computer Science*, pages 43–52. Springer-Verlag, 2006.

[42] L. Thomas, M. Ratcliffe, and A. Robertson. Code warriors and code-a-phobes: A study in attitude and pair programming. In *Proc. 34th Technical Symp. Computer Science Education (SIGCSE'03)*. ACM, 2003.

[43] A.E.M. Van Vianen and C.K.W. De Dreu. Personality in teams: Its relations to social cohesion, task cohesion, and team performance. *European J. Work and Organizational Psychology*, 10:97–120, 2001.

[44] K. Visram. Extreme programming: Pair-programmers, team players or future leaders? In *Proc. Eighth IASTED Int'l Conf. Software Engineering and Applications*, pages 659–664. Acta Press, 2004.

[45] A. von Mayrhauser and S. Lang. A coding scheme to support systematic analysis of software comprehension. *IEEE Trans. Software Eng.*, 25(4):526–540, July/Aug. 1999.

[46] A. von Mayrhauser and A.M. Vans. Industrial experience with an integrated code comprehension model. *Software Eng. J.*, pages 171–182, Sept. 1995.

[47] L. Williams and R.R. Kessler. *Pair Programming Illuminated*. Addison-Wesley, 2002.

[48] L. Williams, R.R. Kessler, W. Cunningham, and R. Jeffries. Strengthening the case for pair programming. *IEEE Software*, 17(4):19–25, 2000.

[49] L. Williams, L. Layman, J. Osborne, and N. Katira. Examining the compatibility of student pair programmers. In *Proc. AGILE 2006*. IEEE Computer Society, 2006.