Do moods affect programmers' debug performance?

Iftikhar Ahmed Khan · Willem-Paul Brinkman · Robert M. Hierons

Abstract There is much research that shows people's mood can affect their activities. This paper argues that this also applies to programmers, especially their debugging. Literature-based framework is presented linking programming with various cognitive activities as well as linking cognitive activities with moods. Further, the effect of mood on debugging was tested in two experiments. In the first experiment, programmers ($n = 72$) saw short movie clips selected for their ability to provoke specific moods. Afterward, they completed a debugging test. Results showed the video clips had a significant effect on programmers' debugging performance; especially, there was a significant difference after watching low- and high-arousal-evoking video clips. In the second experiment, programmers' mood was manipulated by asking participants ($n = 19$) to dry run algorithms for at least 16 min. They performed some physical exercises before continuing dry running algorithms again. The results showed a significant increase in arousal and valence that coincided with an improvement in programmers' task performance after the physical exercises. Together, this suggests that programmers' moods influence some programming tasks such as debugging.

I. A. Khan (✉)
University of Engineering and Technology, Peshawar, Pakistan
e-mail: iftikharahmed@nwfpuet.edu.pk

W.-P. Brinkman
Delft University of Technology, Delft, The Netherlands
e-mail: w.p.brinkman@tudelft.nl

R. M. Hierons
Brunel University, London, UK
e-mail: rob.hierons@brunel.ac.uk

1 Introduction

About 27% of system failures in various companies are being because of software defects according to a survey conducted by Gartner Data quest (Rocco and Igou 2001). Errors in software result in low performance of application and are an indication of poor performance of programmers. There are various reasons that can result in low performances (Smith and Keil 2003), including psychological causes. Attention deficits and lapses in attention are well known to cause a decline in work performance (Coetzer and Richmond 2007; Cheyne et al. 2006), personal productivity, and the quality of life as well as are the cause of accidents (Cheyne et al. 2006). Cognition studies traditionally deal with concepts such as reasoning, perception, intelligence, learning, and various others properties that describe numerous capabilities of the human mind. Cognitive functions also include moods, emotions and feelings (Schmitt 1969; Izard et al. 1984). This poses therefore the question of whether mood might also affect the performance of programmers. Literature reports that moods affect various different activities of people like creativity (Russ and Kaugars 2001; Kaufmann 2003), memory tasks (Lewis and Critchley 2003), reasoning (Chang and Wilson 2004), behavior (Kirchsteiger et al. 2006), cognitive processing (Rusting 1998), information processing (Armitage et al. 1999), learning (Weiss 2000; Ingleton 1999), decision-making (Gardner and Hill 1990; Kirchsteiger et al. 2006) and performance (Chang and Wilson 2004; Lowther and Lane 2006; Lane et al. 2005). Also, Affective Event Theory presented by Weiss and Cropanzano (1996)

establishes that organizational events can cause affective reactions (moods, emotions) that in turn influence job performance and satisfaction. It is evident from the literature that moods influence people's activities and performance. However, there is very limited research on the impact moods might have on information technology professionals (Shaw 2004). A better understanding of this might help to improve the performance of programmers and by doing so reduce software defects.

Programmers create software. Their task is to transform general informal understanding of a goal into a formal model using operators and components so that it is interpretable by the computer (Fisher 1987). This transformed task is what people normally refer to as programming. Programming is one of the tasks of the Software Development Life Cycle (SDLC) among various different tasks like analysis, design and implementation. Programmers like other humans make errors and may therefore introduce errors in software applications. Absentmindedness and attention diversions like distractibility, poor selective attention and mental errors can all be caused by cognitive failures (Larson et al. 1997; Reason 1995). Several studies (Cheyne et al. 2006; Danaher 1980) have shown that cognitive aspects affect attention and the human focus. According to a survey conducted in 2006 by PK500[1], 86% people believe that their performance at work is related to their mood (Pearnkandola 2007). Sad feelings at someone's death or joyful and happy feelings on the birth of a child or at a wedding are the example of moods. Similarly, in almost every task from flying a plane to performing an operation or programming a real-time system, one would expect to feel a wide range of positive and negative feelings (Martin 2001). Researchers for example Ekman (2003) argued that moods affect thinking, e.g., it narrows alternatives people tend to consider. He continues that mood can make it difficult to perform tasks, and may affect performance indirectly while someone is at work, as moods make it difficult to control what we do. These disruptions with interference in energy, sleep, and thinking are common, painful, and too often fatal (Jamison 2003). On the other hand, relaxation, happiness and mood pleasantness can increase someone's creativity (Norman 2004). In other words, programmers in a less pleasant mood might be less able to consider software problems from different angles and therefore less able to find a viable solution.

The above section illustrates that moods might have an impact on programmers' performance, the main position taken in this paper. The following section will elaborate on this, by first establishing a framework using the literature. This framework gives indirect support for a link between programmers' mood and their performance. After this, two empirical experiments are presented showing a direct link between moods and programmers' performance. As one of the many activities of programmers, the experiments concentrated on debugging as a process that especially focuses on removing error in the software. Besides improving understanding about human factors that might affect programmers' performance, this study also makes a more practical contribution as it suggests how programmers can improve their performance. This study also shows how software support tools can be designed to evoke moods that could enhance performance. These scientific contributions and practical implications will be discussed in more detail at the end of the paper.

2 Moods and emotions

Kleinginna and Kleinginna (1981) in a review on the definitions of emotion explained difficulties in defining emotions but agreed on most of the Plutchik's definitions for emotions. Plutchik (1980) defined emotions as states that are aroused by external stimuli and are directed toward the particular stimulus in the environment by which they are aroused. He further added that emotions may be, but are not necessarily or usually, activated by a psychological state. There are no 'natural' objects in the environment toward which emotional expression is directed, and an emotional state is induced after an object is seen or evaluated, and not before.

The definition of emotion from Plutchick is being used quite extensively in most of the latest research, e.g., Gendolla (2000), Cowie et al. (2001) and Izard (2007). According to Parkinson et al. (1996, p. 4), both moods and emotions are considered as affective states as they defined moods as "Affects that refer to mental states involving evaluative feelings". In other words, moods are psychological conditions in which the person feels good or bad, and either likes or dislikes what is happening. Moods are assumed to be affective states that tend to last a little longer and are weaker states of uncertain origin, whereas emotions last for a short time, are more intense, and have a clear object or cause (Fridja 1993). Moods may last from a few minutes to days, whereas emotions come and go in minutes and sometimes even in seconds (Ekman 2003). Most psychologists consider mood and emotion as the same entity. For example, Dow (1992) considers mood as a type of emotion. Most studies also use the terms mood and emotions interchangeably without differentiating between them (Kirchsteiger et al. 2006; Kaufmann and Vosburg 1997). According to Ekman (2003), mood is a continuous emotional state, whereby moods differ from emotions on the basis of the time they are retained and on the variation in the intensity. Furthermore, given an emotion, it is possible

---

[1] A quarterly survey of business and human resource professionals.

to identify the events that caused it; however, the event responsible for causing a mood is rarely known. Moods can just happen; for example, one can wake up in morning with some particular mood. Zimmermann et al. (2003) stated that moods might result in preconceived emotions, as people may not be aware of their moods until attention is drawn toward their mood.

Besides thinking of moods as discrete states (Ekman 2003), mood is also often described as two-dimensional having with dimensions: valence and arousal (Lang 1995; Russell and Barrett 1999). Valence is defined as the degree of happiness or sadness, whereas arousal is defined as a subjective state of feeling activated or deactivated (Sanchez et al. 2005). Figure 1 explains some affective states and their correspondence to some valence–arousal combinations. For example, the "Rejoice" state could be mapped as very pleasant and high aroused, whereas "gloomy" could be mapped as unpleasant and low aroused. Similarly "terrified" could be mapped as unpleasant but high aroused, whereas "soothing" could be mapped as pleasant but low aroused (Sanchez et al. 2005).

Although there are researchers who believe moods to be discrete in structure and trace it back to Charles Darwin's theories (Darwin 1965), there are various other researchers who consider moods as dimensional (Dienstbier 1984). An in-depth discussion about discrete versus dimensional moods is provided by Remington and Fabrigar (1995). Still, the use of two-dimensional models to examine mood is gaining increasing acceptance (Thayer et al. 1994). Various researchers consider mood as, at least, a two-dimensional construct, with an evaluation (valence or pleasure) dimension and an arousal dimension. For example, Almagor and Ben-Porath (1989) divided two dimensions as arousal/activation dimension and hedonic tone/pleasantness dimension. Others like Morris (1995) prefer
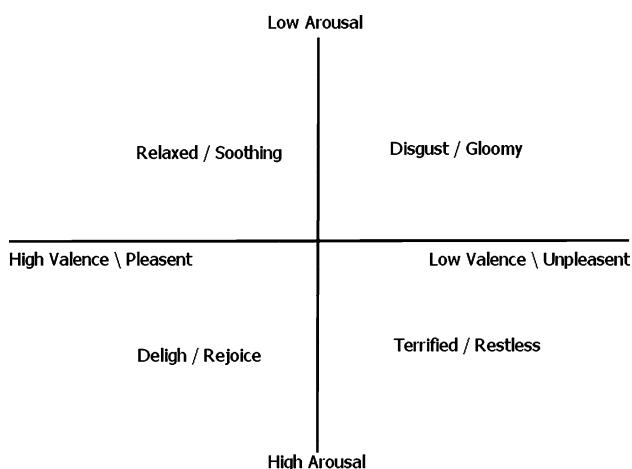
an additional third dimension to express the dominance (controlled, un-controlled). This study used only a bi-dimensional model, with the two dimensions: valence and arousal as these two dimensions are primary and are used in most emotional judgments (Bradley and Lang 1994).

Besides differentiating between moods and emotions, another issue is how mood variations might be observed on a daily basis. Further, several researchers have considered way to change people's mood. Robelin and Rogers (1998) found a significant effect of coffee on energetic mood as well as on psychometric performance. Shepard (1997) found that the learning ability of children, in their curricular time, increased after substantial physical activity. He also found physical activity to be related with increased cerebral blood flow, higher arousal changes in hormone level, changes in body built, etc. Jarvis (1993) found significant improvements in simple reaction time, choice reaction time, incidental verbal memory and Visio-spatial reasoning after the intake of coffee. Similarly, Smith et al. (1993) found significant effects of intake of coffee on alertness and performance. Various studies also targeted the use of a computer in mood regulation and the improvement of performance. For example, Silvestrini and Gendolla (2007) used computer to display film on the computer screen to change the mood of participants.

3 Programming, cognition and mood

Moods and emotions are considered to be cognitive (Schmitt 1969; Izard et al. 1984). Cognitive activities are important for different programming tasks. For example, Schneiderman and Mayer (1979) related program comprehension with memorization. Iron (1982) found induction and syllogistic reasoning to be related. He also found a strong relationship of induction with program composition and its production. Next, Iron also found that spatial scanning and debugging are interrelated. Similarly, Lui and Chan (2004) related learning and memorization with efficient task processing as well as a means for the production of various solutions in repeated tasks. Literature[2] suggests that Programming Tasks (PTs) are related with Cognitive Tasks (CTs) and so these can be mapped to each other in a two-dimensional framework as illustrated below in Table 1.[3]

Several reports in the literature suggest that moods can influence cognitive processes. For example with respect of

Fig. 1 Valence–arousal model with some examples of moods/emotions

---

[2] The nature of literature utilized was mainly qualitative making information on statistic parameters not possible.

[3] This table does not present an exhaustive overview of the literature, instead it just provide some examples from the literature to demonstrate a relation between programming tasks and cognitive functions. A exhaustive meta-analysis paper might be a possible future research.

Table 1 A two-dimensional PCT (Programming Cognitive Tasks) framework

Some cells show examples from the literature that address the specific cross-section of programming task and cognitive element

Programming tasks					
Maintenance	Mayrhausar (1994)				
Debugging	Iron (1982)		Iron (1982)		Iron (1982)
Program Construction /Modelling	Mayrhausar (1994)		Iron (1982)	Iron (1982)	
Program comprehension	Lui and Chan (2004)	Lui and Chan (2004) Iron (1982)			
Definition	Lui and Chan (2004)				Iron (1982)
	Memory	**Closure**	**Induction**	**Reasoning**	**Spatial scanning**	*Cognitive tasks*

Table 2 A Moods Cognitive Task (CMT) framework, showing moods (valence, arousal) impact on cognitive tasks (CTs)

* Indicates that a specific study does not focus exclusively on only valence or only arousal, but on mood in general

Moods					
Valence	MacWhinney et al. (1982) Lewis and Critchely (2003)	Zheng (2008)* Delplanque et al. (2005)*	Zheng (2008)* Delplanque et al. (2005)*	Chang and Wilson (2004)	Zheng (2008)* Delplanque et al. (2005)*
Arousal	MacWhinney et al. (1982) Lewis and Critchely (2003)	Bartolic et al. (1999)* Roets and Hiel (2008)	Bartolic et al. (1999)*	Chang and Wilson (2004) Mujica-Parodi et al. (2004)	Bartolic et al.(1999)*
	Memory	**Closure**	**Induction**	**Reasoning**	**Spatial scanning**	*Cognitive tasks*

human memory, MacWhinney et al. (1982) as well as Lewis and Critchley (2003) found that the mood dimensions valence and arousal are related to memory performance. Zheng (2008) also argued with the help of literature that valence and arousal are related to implicit and explicit memory, whereas Mujica-Parodi et al. (2004) found that arousal had a significant impact on cognitive performance. Chang and Wilson (2004) found that valence and arousal affect reasoning capabilities. Based on this literature, it seems that moods (valence, arousal) and cognitive tasks (CTs) are related and can be mapped onto each other in a two-dimensional framework as illustrated in Table 2.

The literature provides direct support for a link between programming task and cognitive processes (Table 1) and for a link between mood and cognitive process (Table 2). Logically, taking these two together provides indirect support for a link between the programming task and mood. Thus, another two-dimensional framework with one axis representing moods (valence, arousal) and the other axis representing programming tasks can be formulated as represented in Fig. 2. This supports the hypothesis put forward in this paper. The symbols in the cells indicate a
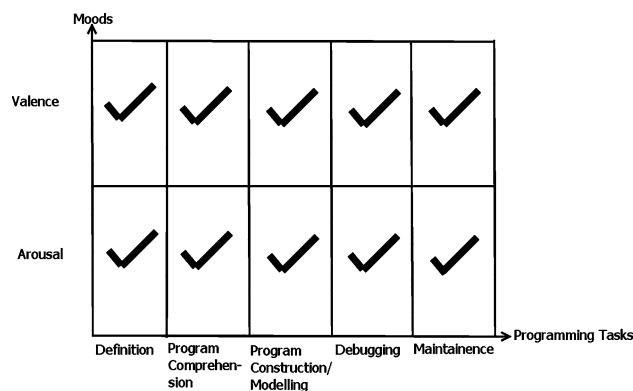

Fig. 2 A framework based on frameworks described in Tables 1, 2 showing moods might have an impact on Programming Tasks (PT's). *Ticks* in the respective cell represent the relations

possible relation between specific programming task and the specific mood dimension. For example, Lewis and Critchley (2003) presented a case to show that valence (positive or negative) has an effect on memory. Similarly, Schneiderman and Mayer (1979) found that memory plays

a vital role in programming tasks like definition (D), program comprehension and understanding (PC), program construction and modeling (PM) and debugging (DG). As valence affects memory and memory affects programming tasks (D, PC, PM, DG), there might be a relation between the programming tasks (D, PC, PM, DG) and valence. The framework also shows that all cognitive tasks mentioned here (induction, closure, memory, induction and reasoning) are related with debugging, thus indicating the impact of moods on debugging. Debugging is a task that focuses on finding and correcting errors in a program. Debugging is a difficult task and requires a large part of a programmer's effort (Brooks 1975). It is a complex but important activity and is the combined process of testing and code correction (Shaochun and Rajlich 2004). Debugging has received some attention from software psychology (Rugaber and Tisdale 1992), but there is still a lack of research in the scope of cognitive abilities and debugging performance. Therefore, the main focus of this study will be on debugging.

The literature-based framework provides indirect support for a link between moods and programming task. Still, no direct support was found in the literature. Based on the literature, the study puts forward a relationship between moods and debuggers' performance. Therefore, the main research question for this study is divided into two directional hypotheses:

1. High valence will increase debuggers' performance, while low valence will cause degradation in debuggers' performance.
2. High arousal will increase debuggers performance, while low arousal will decrease debuggers' performance.

The next sections therefore address these hypotheses. Two experiments are presented, one conducted over the Internet, the other in the lab.

4 Study 1

The idea behind the first study was to give programmers a debugging test after they watched mood-inducing movie clips and compare programmers' performance. The experiment involved five movie clips to induce moods and a multiple-choice debugging exercise to measure the programmer's performance on recognizing errors in pieces of software code.

4.1 Methods

The study was conducted over the Internet. Results from web experiments and surveys have been reported to yield the same conclusions as studies done in lab conditions despite great differences between the subject characteristics

(Krantz and Dalal 2000). Conducting the experiment on the web provides additional benefits like (1) Participants are recruited locally and globally and therefore more representative of the population (generalizable demographically). (2) Web studies have high external validity and depict original behavior (Gosling et al. 2004). (3) The experiment can be taken at any time at participant convenience (time generalizeable). (4) The costs are low in web studies. (5) Web studies can give high statistical power by enabling access to larger samples (Krantz and Dalal 2000). In addition, participation of the participants in web experiments is completely voluntary (Reips 2000).

Of course, there are also some disadvantages to this type of experimentation like little control of the environmental conditions and a high dropout rate from Internet experimentation. Despite disadvantages, data gathered using Internet experiments seem trustable because of the high correspondence of results between Internet experiments and laboratory setting experiments. Similarly, a high dropout rate is an indication that participation in the experiment is voluntary (Krantz and Dalal 2000). Given these considerations, conducting the experiment on the web increased the chances of involvement of professional programmers required for the study, which is relatively difficult to obtain in a lab setting.

The experiment was set up with a two-by-two factorial design with valence (low/high) and arousal (low/high) levels as two between-subjects variables. The participants' debugging performance measured in a less extreme, or other words, neutral mood condition was included as a covariate in order to control variance caused by individual difference in debugging skill. Each participant watched a neutral movie clip of one minute duration. This movie clip was expected to induce neutral mood before actual experiment. The dependent variables were participants' performance on debug task, operationally in the number of correct answers and time needed after seeing one of the four mood evoking video clips.

4.1.1 Materials

A total of five movie clips were used to induce moods in this study. Mood induction procedures are widely used in various studies like Kirchsteiger et al. (2006) and Nasoz et al. (2003). Four movie clips represented the four quadrants of two-dimensional (valence, arousal) model with one movie clip selected as neutral condition. These movies and the neutral movie had previously been empirically tested (Khan et al. 2007) on their ability to evoke specific valence and arousal responses.

To measure the debugging performance, two sets of six debug questions were created. Each set contained three easy and three medium questions. These questions were

taken from a textbook on C++ (Deitel et al. 2000). To avoid floor or ceiling effects of the questions too simple or too difficult, three questions were selected as easy that were rated medium by one and easy by two computer science lecturers. Similarly, three questions were selected as medium that were rated as medium by two and difficult by one computer science lecturer. The overall time to complete the debug test was limited in order to ensure that the effect of the movie clips remained throughout the test. Every question had a separate timing, which was allocated based on the text and the number of calculations involved in a question. For example, if a question contained 7 lines of code, then this question was allocated 70 s. A question involving a computation like $(x = 2 + 2 \times 10/2 - 2)$ was allocated 50 s, although it only contained one line of code since it contained five different executable statements as shown in the example below.

$2*10 = 20$	(statement 1)
$20/2 = 10$	(statement 2)
$10 + 2 = 12$	(statement 3)
$12 - 2 = 10$	(statement 4)
$x \leftarrow 10$	(statement 5)

The minimum time allocated to any question was 30 s. If participants answered the question before the time deadline, they were moved automatically to the next question in the test. Total time allowed answering the entire questions set was 250 s. Two example questions for the programming language C/C++ can be found in "Appendix I". Finally, the questions were translated into C#, Java and Visual Basic dot net, allowing programmers to take the debugging tests in their preferred programming languages.

4.1.2 Participants

Data were obtained from 372 cases in which people started the experiment; however, in only 75 cases was the experiment completed with a dropout percentage of 80%. In the start of the experiment, an input item also asked participants to enter the number of times that they had previously done this experiment. Three out of 75 participants indicated that they had done the experiment at least twice. Therefore, the analysis was done on 72 cases in which participants took the experiment for the first time.[4] Five

---

[4] Additional, individual data [age, gender, experience, type of programmer (Under Graduate, Postgraduate, PhD Student, Professional) and language selected to take test in)] was checked for uniqueness and were found unique. Analysis of unique IP addresses was not possible as the server was behind a proxy.

participants were females and 67 were males. Participants' age ranged from 18 to 44 years with a mean of 26.3 and SD of 5.2. Programming experience ranged from 0.5 to 25 years with a mean of 4.8 years and SD of 5.6. Participants included 81% professionals, 8% postgraduates, 6% undergraduates, 4% PhD students and 1% hobby programmers. Of the participants who took the debug test, 58% took it in C/C++, 24% took it in C#, 11% used Java and 7% used Visual basic dot net.

4.1.3 Procedure

The experiment was conducted on the Internet with invitations being sent via various programming forums. At the start of the experiment, participants saw an introduction page, which explained the nature of the experiment and requested the participants to give their consent for participating in the experiment and for the collection of data. The participants went through a training session, so that they should be familiar with the sequence of the test. The training session consisted of watching a neutral mood video clip, followed by a series of debug questions that they had to answer within a fixed time. After completing the training session, participants continued with the actual test, which consisted of two cycles of the movie and debugging test. There were two fixed debugging tests of six questions. The movie clips were either the neutral movie clip or one of the four mood-inducing movie clips. To control for potential training effects, the order of the neutral movie and the mood-inducing movie clip as well as the selection of the specific mood movie clip was systematically controlled. The system assigned participants in the cycles of eight to the eight possible sequences (four clips, two orders for each). Furthermore, the assignment of the two versions of the debug test to the first or the second cycle was counter balanced.

4.2 Results

The first preparation step toward the analysis was to examine a potential self-selection bias between the conditions of the experiment, e.g., less motivated people that drop out of the experiment after seeing the low arousing movie, leaving only highly motivated people in that specific condition. Especially for web experimentation, with its truly voluntary nature, it is important to examine the distribution of the dropout rates across the experimental conditions (Reips 2000). A systematic difference between the conditions would confound the interpretation of any causal effect between the experimental conditions and the programmers' performance.

A total of 72 (20%) participants completed the study, and rest of 80% decided to drop out. An analysis on dropout data showed that 93% participants decided not to

Table 3 The number of participants that complete the test, number of dropouts and dropout rate of participants dropping out after being allocation to experimental conditions

Valence	Arousal								
	Low			High			Total		
	N.C[a]	N.D[b]	P.D[c] (%)	N.C[a]	N.D[b]	P.D[c] (%)	N.C[a]	N.D[b]	P.D[c] (%)
Low	7	1	13	14	5	26	21	6	22
High	31	7	18	20	4	17	51	11	18
Total	38	8	17	34	9	21	72	17	19

[a] Number of participant completed experiment

[b] Number of participant dropout after being allocated to condition

[c] Percentage of participant that dropout after being allocated to condition

participate after reading the project description, while only 7% decided to terminate participation half way in the experiment when they were assigned to an experimental condition. Table 3 shows the distribution across the experimental conditions of the 7% dropout and the number of people that participated. A Fisher's exact test[5] on this 7% dropout rate revealed no significant ($p > 0.05$) variation between the experimental conditions, and therefore canceling out self-selection as an alternative explanation for any causal effect found. Still examination of Table 3 did reveal a significant (χ^2 (3, $N = 72$) = 17.2, $p = 0.001$) deviation from an equal distribution of participant that allocated to specific condition and completed the experiment. Although this does not confound any potential statistical results, it constrains possible statistical analyses such as a full orthogonal factorial 2×2 analysis.

A number of Multivariate Analysis of Covariance were conducted to examine the effect the videos had on the performance of the debug task. The individuals' ability to complete these debug tasks was controlled by looking at performance after seeing the neutral movie, i.e., the neutral mood condition. The covariate was therefore the number of correct answers in neutral mood condition. As mentioned before, the numbers of participants from the quadrant low valence low arousal (LVLA) were too low to conduct a 2×2 analysis. Therefore a Multivariate Analysis of Covariance was carried out on only the movies from the other three quadrants (HVHA, HVLA, and LVHA), creating therefore a design for the analysis with only one fixed independent variable, with 3 levels for the movies, besides the covariate.

The two dependent variables were the number of correct answers and the number of tasks completed within the time set. To control for a potential inflated Type I error rate, the first step was to conduct an overall multivariate analysis to examine whether a significant main effect could be found across the two dependent variables. Only if this was the case, univariate analyses on the individual measures were considered using a Bonferroni correction for per comparison alpha level.

The multivariate analysis results showed a significant main effect (F (2, 65) = 3.54, $p = 0.036$) for the videos. Examining the results of the univariate analyses showed that this effect could be found in the number of correct answers (F (3, 65) = 6.35, $p = 0.001$) and in the number of tasks completed within the time set (F (3, 65) = 5.18, $p = 0.003$), which were both significant for a Bonferroni correction of a per comparison alpha level of 0.025, for a family wise alpha of 0.05. These results suggest that mood could affect participants' performance in the debug tasks.

The next step was to examine this effect on both the arousal and valence dimension. To study the arousal dimension, the previous analyses were repeated. However, this time the only fixed independent variable movie had only two levels (high valence/low arousal HVLA and high valence/high arousal HVHA). In other words, valence was kept constant at the high level condition, while arousal level varied. The results of the multivariate analysis showed a significant effect for the movies (F (2, 49) = 5.12, $p = 0.01$), and when applying a Bonferroni correction of alpha 0.025, this main effect was only found in univariate analysis on the number of tasks completed within time (F (1, 50) = 10.01, $p = 0.003$), but not on the number of correct answers (F (1, 50) = 3.70, $p = 0.06$). Examination of the averages, corrected for the effect of the covariate, showed that the participants completed 3.03 tasks within time set in the low arousal condition and 4.59 in the high arousal condition. This therefore suggests that arousal level can affect programmers' debug performance, and in this experiment, performance was improved with increased arousal level.

Similar analyses were conducted for the valence dimension. This time the independent fixed variable only

included the low valence/high arousal (LVHA) condition and the high valence/high arousal (HVHA) condition; thus, arousal was kept constant at high arousal condition. The results of the multivariate analysis this time found no significant effect for movies (F (2, 33) = 1.86, p = 0.17, η^2 = 0.1). Although this does not suggest that valence will never affect debug performance, it was simply not found in this experiment. The relative small sample size and the effect size of η^2 = 0.1 might have been determining factors[6] in this case. Potential violations of the assumption for the covariance analyses were also tested such as (1) the linearity of the relationship between the covariate and the dependent variable[7] and (2) homogeneity of covariate-dependent variable slopes.[8] No indication of a violation was found.

To conclude, the results of the first study suggests that moods, particularly, arousal affected the programmers' performance. A second study was then developed to confirm the finding, this time invoking a mood change in a different way.

5 Study 2

In this study, participants were under arousal while tracing algorithms. Using an intervention to increase the arousal level allow for studying the effect of this on programmers' performance. There are various strategies to affect people's arousal level. For example, Watters et al. (1997) used caffeine to arouse participants in an experiment to test the validity of Yerkes-Dodsons' Law. Similarly, physical exercises are also known for their positive impact on performance. For example, McMorris and Graydon (1997) found that exercise could significantly increase the speed of visual search, speed of decision-making and accuracy of decision-making. Physical exercises are also known to have an impact on moods. For example, Steptoe and Cox (1988) found that moderate physical exercises result into positive moods. Gowans et al. (2001) after an experiment involving physical exercises concluded that exercises had improved the moods and physical functions of their subjects.

---

[6] Nancy et al. (2005) termed an effect size of η^2 = 0.1 as small effect. With sample size of N = 37 and p > 0.05 the difference in a sample might be attributed to chance alone.

[7] Significant correlations were found between the number of correct answer in the neutral condition (covariant) and in mood condition (dependent) (r (72) = 0.42, p < 0.001) and between the number of correct answers in the neutral condition and number of task completed with in time (dependent) in neutral condition (r (72) = 0.45, p < 0.0001) and in mood condition (r (72) = 0.32, p < 0.01).

[8] No significant interaction effect was found between the covariant and other independent variable (mood).

Therefore, physical exercises were introduced in the second study to manipulate the participants' arousal level.

5.1 Methods

The aim of the experiment was to determine the impact of an intervention in the form of some physical exercises, on the programmers' program understanding and debugging performance. Therefore, a total of 24 algorithms were selected and divided into three categories of easy, medium and difficult. Different levels of difficulty were selected to ensure that programmers with different programming skill could participate in the experiment. The variation in difficulty would reduce potential floor or ceiling effects in programmers' performance.

The aim of the experiment was to decrease the arousal level of participants over time by asking them to trace various algorithms for a period of at least 16 min. Some degree of boredom would set in causing a low level of arousal. After 16 min, an intervention was introduced. The intervention was in the form of a video in which participants were asked to participate in some simple warming up exercise. After the intervention, the participants continued with tracing algorithms for about another 8 min. Analyzing the performance before and after intervention provided an insight into the impact of the computer-based mood changing intervention on participants' performance. The design of this study might introduce task-induced fatigue on the participants. This study was designed to induce boredom or sleepiness; therefore, task-induced fatigue and low arousal might not be different. Researchers like Desmond and Matthews (2001) showed that fatigue proneness is negatively associated with energetic arousal and may be cause of the boredom. This means that an increase in fatigue can cause a low arousal level or sleepiness.

5.1.1 Materials

The algorithms were selected mainly from Parberry and Gasarch (2002) along with a basic algorithm book "Data structures and Algorithms in Java" from Lafore (1998). Parberry and Gasarch (2002) classified their algorithms into three difficulty levels. The algorithms taken from Lafore (1998) were categorized into three difficulty levels according to the type of tracing and reasoning involved and the data structures used in the algorithm. For example, an algorithm with a single or nested loop was categorized as 'easy' if it involved simple tracing and if no complex computations were involved. An example of an easy algorithm is one that contained simple loops and if-else structures. A medium algorithm was a mixture of loops with some basic reasoning and logic required in order to create the trace table. An example of a medium algorithm

is one that contained several nested loops or nested if-else structures in addition to some mathematical computations that increased the complexity of algorithm and tracing steps. A difficult algorithm was composed of algorithms with un-orthodox looping styles like recursion. These algorithms also contained some complex data structures, which might prove to be difficult to trace. An example of an algorithm for each level of complexity can be found in "Appendix II".

For the mood changing intervention, a special exercise video was prepared. The video had playtime of 2 min and 17 s. It contained some very simple exercises like warming up by moving hands, legs and jumping. Background music was played with the exercise instruction video. Furthermore, to validate that the actual mood change had occurred, participants were asked to rate their valence and arousal level on Self-Assessment Manikin (SAM) scale (Lang 1980). The scale ranged for valence from 1 being happy to 9 being sad. Similarly, arousal ranged from 1 being excited to 9 being calm and sleepy.

5.1.2 Participants

Invitations to the participants were sent via email. Participants were also approached via personal contacts. A total of 19 participants participated in the study. The mean age of participants was 28.1 years with a standard deviation of 4.5. There was only one female participant. A total of 79% of the participants categorized themselves as programmers; 16% categorized themselves as expert computer users and 5% categorized themselves as medium computer users. The mean programming experience of the participants was 8.3 years with a standard deviation of 2.9 years.

5.1.3 Procedure

The experiment was a controlled experiment with no one going in or out of the room during the experiment. All other distracting devices like mobiles phones were also switched off. The experiment started with a training session. Algorithms appeared on the screen in proper indentation and formatting to make them easy to read. Participants were required to produce a trace table for each algorithm. Participants could write their answers in a separate text box in the application. Participants had four minutes to complete each algorithm. Participants were able to go to the next algorithm by clicking the 'Next button' if they wanted to or if they completed tracing the algorithm. If participants were not able to complete the algorithm within the required time, any input in the answer text box was automatically saved in the database and participants were presented with the next algorithm. Algorithms kept on appearing for about 16 min. Participants were asked to rate

their mood on the self-reporting two-dimensional SAM scale after 16 min.

After rating their mood, a video was displayed on the participant's screen. Participants were asked to repeat some simple physical exercises as shown in the video. The exercise was followed by another mood rating dialog box. This was followed again by the next sequence of algorithms. Algorithms always appeared in a cycle of easy, medium and difficult before the intervention. However, after intervention, this order was reversed. For example, a participant answering an easy, a medium and a difficult algorithm would receive after the intervention the sequence difficult, medium and easy. This arrangement ensured that performance before and after the intervention was balanced as the level of difficulty was spread equally.

5.2 Results

Two markers marked 147 algorithm traces from the 19 participants on two criteria: correct output and correct flow. Pearson correlation between the markings of the two markers indicated high level of consistency (correct output: $r(146) = 0.88$, $p < 0.0001$; correct flow: $r(146) = 0.72$, $p < 0.0001$). The averages of the markings of the both markers were therefore taken as final markings for these two measures. The other performance measures used were 'time left to complete tracing', 'total number of correct variables identified' and 'total lines of correct output'. These measures were automatically calculated ensuring consistency in the marking.

A paired samples t-test conducted on arousal rating immediately before and after the intervention indicated that the participants' arousal level was significantly higher ($t(18) = 6.7$, $p < 0.0001$) after the exercises ($M = 4.4$, $SD = 1.9$) than before it ($M = 6.3$, $SD = 1.6$). Similar t-tests conducted on valence rating also showed that valence level was significantly more positive ($t(18) = 6.9$, $p < 0.0001$) after the exercises ($M = 4.2$, $SD = 1.8$) than before exercises($M = 6.0$, $SD = 1.6$). These findings suggest that the physical exercises had a significant impact on the participants' mood—the mediating factor that was expected to influence the participants' performance.

To examine the effect of the intervention on performance, a repeated multivariate analysis was conducted on the measures: 'total correct variables identified', 'total correct lines of output', 'correct flow', 'correct output' and 'time left' just before and after intervention, which was the only independent within-subjects variable in this analysis. Results showed that participants' performance improved significantly after the intervention ($F(5, 19) = 3.51$, $p = 0.03$). Follow-up univariate analyses on the individual measures only revealed a significant effect in the correct output ($F(1, 19) = 13.81$, $p = 0.002$) measure, which is
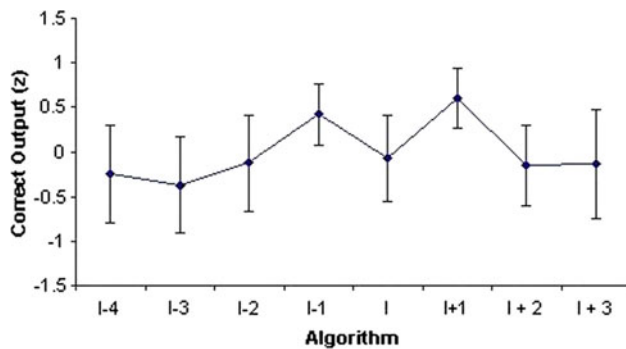
Fig. 3 Correct output performances standardized by difficulty level, where I stand for the algorithm complete just before the intervention, and $(I + 1)$ is the first algorithm completed just after the intervention. Note that algorithms $(I - 3)$ to $(I + 2)$ were complete by all 19 participants, algorithm $(I - 4)$ by only 17 and $(I + 3)$ by only 12 participants

significant for a Bonferroni correction (alpha = 0.01). Figure 3 shows the mean marks given to correct output of the algorithms just completed before (I) and after $(I + 1)$ the intervention. The marks are presented in z-values. The mark for the correct output was slightly below the average mark (-0.07) just before the intervention (I), whereas participants got the highest mark of 0.60 SD above the average marking for the first algorithm complete just after the intervention $(I + 1)$. Looking at Fig. 3, it seems that up to $I - 1$ performance was still improving as part of learning effect, and after this point, possible fatigue or boredom might have set in. The effect of the intervention also seems temporary as performance seems again to drop in the second $(I + 2)$ and the third $(I + 3)$ completed algorithm after the intervention.

The findings of the study suggest that an increase in arousal and/or valence after computer-based intervention in the form of physical exercise coincided with an increase in the performance in the algorithm-tracing task. This is an effect that could not simply be explained by learning effect over time, as a decline in performance seems to have set in just before the computer-generated intervention. Still, another limitation could be the time pressure as participants had to complete the algorithms within 4 min. However, time pressure is not unrealistic for an industrial environment.

In theory, the observed performance improvement could also be attributed to a reduction in fatigue caused by the physical exercise or a temporally change of the mental demands. Still, this would have been coincided with a change in the participants' mood.

6 Conclusion, limitations and future research

Both the Internet study and the controlled lab study showed that mood could have an effect on programmers'

debugging performance. Although this effect was indirectly supported by literature, the scientific contribution of this works is to demonstrate this hypothesized effect directly in empirical settings. An additional contribution is the presented framework as it can lead future research to study the effect of mood on other programming tasks besides debugging such as program comprehension or program modeling. Enhanced insight could lead to software support tools for programmers that would take into consideration this mood effect. A mood aware development environment could use mood information to help programmers to regulate their mood and enhance their performance. The second study already demonstrated intervention delivered by a computer to be effective. Besides using self-reported moods instrument, this environment should also consider using other methods for measuring mood, for example, physiological measures (e.g., heart rate, perspiration) or behavioral measures (e.g., keyboard use) (Khan et al. 2008). Besides monitoring mood levels, the program might also be able to monitor performance level to decide when to suggest a mood changing intervention. In addition, research on the presentation of the interventions, their usability, their social acceptability and side effects also need to be studied in detail in order to implement a development environment that could really help programmers to improve their performance in the context of their moods.

As the high arousal conditions always coincided with the high performance level in the studies, a logical practical implication would be to aim for the high levels of arousal. However, considering the Yerkes–Dodson Law (Yerkes and Dodson 1908), too much arousal might again have a negative impact on the performance. This law suggests an inverted-U shape relationship between arousal and performance. Figure 4 shows that in the two studies, the low arousal condition might have been on the left side of this optimum and the high arousal condition slightly more to the right. Increasing the arousal level even further might
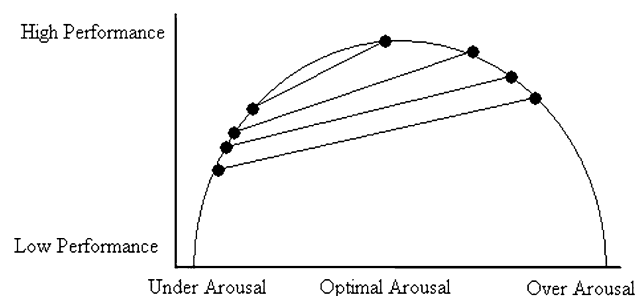

Fig. 4 Arousal performance relationships in terms of Inverted-U shaped hypothesis including the potential places of experimental conditions (*left side* represents low arousal, while the *right side* represents high arousal)

again lead to a drop in performance possibly even below the low arousal conditions. This, however, was not examined in this experiment and therefore would require future research.

Various reports in the literature suggest that valence (positive and negative moods/happiness or sadness) does have an impact on performance. Unfortunately, this could not directly be concluded for the debugging task in the two studies. The first study found no significant effect, and in the second study, the effects of arousal and valence could not be separated. One of the reason for this could be that vigilance/attentiveness is more associated with arousal than valence. Researchers like Isen (2008) indicated that negative affect (low arousal and anxiety) narrows the focus of attention and therefore can degrade performance. Dickman (2000), referring to attentional-fixity theory, indicated that high arousal is related with better performance in attentional-fixity conditions, whereas low arousal is related with degradation in performance. Therefore, the impact of valence on programmers' debug task remains an interesting open question, which future research might be able to answer. As emotions are more intense than being in a certain mood, possible future research could also be to measure the impact of emotions on the debuggers' performance.

Like any empirical study, these studies also have a number of limitations. For example, in exercising Internet-based data collection, it is difficult to make sure that participants should properly be exposed to the experimental conditions. It was difficult to know whether participants actually watched the mood-inducing video clips or not, which in turn might have affected the experiment outcomes. Another limitation is that the task given to the programmers might not be representative of the entire industrial debugging task where most of the efforts are used to locate and identify relevant code while ensuring that these changes did not create any ripple effect. Besides studying the effect of valence and arousal on performance in lab and online, it is also important to study them in an industrial environment where programmers might be more attentive to avoid risks of introducing bugs into the software (Isen 2008).

To our knowledge, no work has been published about risk and its affect on the attention of debuggers or IT personnel. However, various researchers discussed the affects of risks on other types of work like driving. For example, Vaa (2007) discussed drivers' risk compensation as an unconscious behavior, e.g., if a certain risk-reducing measure is introduced in the road traffic system, the expected reduced risk is compensated by certain behavioral changes like increase in driving speed (Vaa 2007). People

in mild positive affect when are in a high-risk situation have more thoughts about losing and therefore behave more conservatively to avoid loss (Isen 2008). Thus, when debuggers feel that there is a hight risk of bugs in the software, they might be more attentive. However, Isen (2008) also indicated that people often pay less attention to the task which are boring and are not profitable or are not a cause of loss to them. Therefore, experiments in this study might be less externally validate-able, as the tasks in these studies may not be of benefit to the participants or cause of loss to them.

Damasio (1994) divided emotions and feeling into three levels: (1) Primary emotions, (2) Secondary emotions and (3) Feelings. He considered primary emotions as innate, unconscious and predominantly corresponding to infants. He defined secondary emotions as emotions that are learnt with experience and develop into 'the emotion of adult'. They are also unconscious or preconscious. Feelings were defined as a process of bringing emotions to conscious. As this study was conducted on adults, secondary emotions are of concern here. As feelings are defined as a conscious process and can be reported at a given time, they might be an equivalent of moods in this study as participants were consciously aware of their moods and therefore rated their moods on mood rating scale. Primary and secondary emotions are unconscious processes that can be represented by SCR (Skin Conduction Response) Bechara et al. (1997). SCR can be utilized to measure moods and emotions unconsciously which can also reduce the bias of the participants while rating the mood scale. Therefore, there is a need for future research that could measure emotions through the use of SCR and their effect on debuggers' performance. Various studies like Khan et al. (2008) used GSR (Galvanic Skin Response) to measure moods and found significant correlations with the keyboard and mouse use.

The findings presented here could be regarded as a first step in developing a deeper understanding, as the experiments show that moods have at least an impact on programmers' debugging tasks. For the moment, on a more practical level, the findings suggest that programmers when their arousal level drops could consider doing some simple physical exercises, as for the short term, this seems to have a positive effect on their debugging performance.

Appendix I

Example debugging questions of study 1

Question: which of the following C/C++ statements contain variables whose values are replaced? (Difficulty Type = Easy)

    a) int b, c, d, e, f, I, j, k, p;
    b) cin ≫ b ≫ c ≫ d ≫ e ≫ f;
    c) p = i + j + k + 7;
    d) cout ≪ "variables whose values are destroyed";
    e) cout ≪ "a = 5";

Answers
1. Statements (b) and (c). (Correct answer)
2. Statements (c) and (d)
3. Statements (d) and (a)
4. Statements (c) and (a)

Question: what is the error (logical, syntax) in the following? (Difficulty Type = Easy)

```
Int                                              main()
{
float sum;
int n = 1, count = 0;
while (n > 0)
{
scanf("%d", &n);
sum = sum + n;
count ++;
}
return(0);
}
```

Answers

1. n is not initialized
2. **sum is not initialized (Correct answer)**
3. count++ will never execute

Appendix II

Algorithms examples from study 2
Algorithm: (Difficulty Level: **Easy)**

Consider the following piece of code

1.	Base := Some Number
2.	Exponent := Some Number

3.	Temp := Nothing	
4.	for i := 1 to Exponent	
5.		Temp = Temp * Base
6.		Output Temp
7.	End for loop	
8.	Output Temp	

Requirement:

Suppose Exponent has a value of 6 and Base has value of 5. Identify all the variables and create a trace table from the algorithm above.

Algorithm: (Difficulty Level: **Medium)**

Consider the algorithm below

1.	i := 0, j := 0, v := 1, N = 4		
2.	for i := 1 to N		
3		for j := 1 to N − i	
4			Output " "
5		End for loop j	
6		for k := j to j + v	
7			Output "*"
8		End for loop k	
9		v := v + 2	
10		Change Line	
11	End for loop i		

Requirement:

Identify all the variables and create a trace table from the algorithm above. Suppose N = 7.

Algorithm: (Difficulty Level: **Difficult)**

Consider the algorithm below

1	Function RecursiveFibonacci(N)
2	if (n = 0) or (n = 1)
3.	return n
4.	else
5.	return RecursiveFibonacci (N-1) + RecursiveFibonacci (N-2)
6.	End If
7.	End Function

Requirement: Identify all the variables and create a trace table from the algorithm above. Assume N = 8.

References

Almagor M, Ben-Porath YS (1989) The two-factor model of self reported mood: a cross cultural replication. J Personal Assess 53:10–21

Armitage CJ, Conner M, Norman P (1999) Differential effects of mood on information processing: evidence from the theories of reasoned action and planned behavior. Eur J Social Psychol 29(4):419–433

Bartolic EI, Basso MR, Schefft BK, Glauser T, Titanic-Schefft M (1999) Effects of experimentally induced emotional states on frontal lobe cognitive task performance. Neuropsychologia 37:677–683

Bechara A, Damasio H, Tranel D, Damasio AR (1997) Deciding advantageously before knowing the advantageous strategy. Science 275:1293–1295

Bradley MM, Lang PJ (1994) Measuring emotion: the self-assessment Manikin and the semantic differential. J Behav Therapy Exp Psychiatr 25:49–59

Brooks FP (1975) The mythical man month. Addison-Wesley, Reading

Chang A, Wilson M (2004) Recalling emotional experiences affects performance on reasoning problems. Evol Human Behav 25:267–276

Cheyne JA, Carriere JSA, Smilek D (2006) Absent-mindedness: lapses of conscious awareness and everyday cognitive failures. Conscious Cogn 15:578–592

Coetzer GH, Richmond L (2007) An empirical analysis of the relationship between adult attention deficit and effi-cacy for working in teams. Team Performance Manag 13(½): 5–20

Cowie R, Douglas-Cowie E, Tsapatsoulis N, Votsis G, Kollias S, Fellenz W, Taylor G (2001) Emotion recognition in human-computer interaction. IEEE Signal Process Magazine 18(1): 32–80

Damasio AR (1994) Descartes' error: emotion, reason and the human brain. G.P. Putnam's & Sons, New York

Danaher JW (1980) Human error in ATC system operations. Hum Factors 22:535–545

Darwin C (1872–1965). The expression of emotion in man and animals. University of Chicago Press, Chicago

Deitel HA, Deitel PJ, Neito T (2000) Instructor manual for C++ how to program, 3rd edn. Deitel and Deitel Associates Inc, USA

Delplanque L, Silver L, Hot P, Rigoulot S, Sequeira H (2006) Arousal and valence effects on event-related P3A and P3B during emotional categorization. Int J Psychophysiol 60(3):315–322

Desmond PA, Matthews G (2002) Task-induced fatigue states and simulated driving performance. J Exp Psychol 55(2):659–686

Dickman SJ (2000) Impulsivity, arousal and attention, personality and individual differences. Person Individ Differ 28:563–581

Dienstbier RA (1984) The role of emotion in moral socialization. In: Izard CE, Kagan J, Zajonc RB (eds) Emotion, cognition and behavior, 1st edn. Cambridge University Press, USA, p 198

Dow J (1992) External and internal approaches to emotion. Psycoloquy 3(1). Reterived October 2009 from Psycholoquy website http://www.cogsci.ecs.soton.ac.uk/cgi/psyc/newpsy?3.01

Ekman P (2003) Changing what we become emotional about. In: Ekman P (ed) Emotions revealed. Times Book (Henry Holt and company) LLC, New York, pp 67–68

Fisher G (1987) Cognitive view of reuse and redesign. IEEE Software (Special Issue on Reusability) 4(4):60–72

Fridja N (1993) Moods, emotion episodes, and emotions. In: Lewis M, Haviland I (eds). Handbook of emotions. Guildford, New York, pp 381–403

Gardner MP, Hill RP (1990) Consumers' mood states and the decision-making process. Market Lett 1:229–238

Gendolla GHE (2000) On the impact of mood on behaviour: an integrative theory and a review. Rev General Psychol 4(4): 378–408

Gosling SD, Vazire S, Srivastava S, John OP (2004) Should we trust web-based studies. Am Psychol 59(23):93–104

Gowans WE, Dehueck A, Voss S, Silaj A, Abbey SE, Reynolds WJ (2001) Effect of randomised, controlled trial of exercise on mood and physical functions in individual with fibromyalgia. Physiotherapy and Occupational Therapy Departments Toronto hospital

Ingleton C (1999) Emotion in learning: a neglected dynamic. HERDSA Annual International Conference, Melbourne, pp 12–15

Iron DM (1982) Cognitive correlates of programming tasks in novice programmers. Proceedings of the conference on human factors in computing systems, pp 219–222

Isen AM (2008) Some ways in which positive affect influences decision making and problem solving. In: Lewis M, Haviland I (eds) Handbook of emotions, 3rd edn. Guildford, New York, pp 548–573

Izard CE (2007) Basic emotions, natural kinds, emotion schemas, and a new paradigm. Perspect Psychol Sci 2(3):260–280

Izard EC, Kagan J, Zajonc BR (1984) Introduction. In: Izard EC, Kagan J, Zajonc BR (eds) Emotions cognition and behavior, 1st edn. Cambridge University Press, Cambridge

Jarvis MJ (1993) Does caffeine intake enhance absolute levels of cognitive performance? Psychopharmacology 110(1–2):45–52

Jamison KR (2003) Foreword, in mood disorders. In: Power M (eds) A handbook of science and practice (1st edn)

Kaufmann G (2003) Expanding the mood-creativity equation. Creativ Res J 15(2,3):131–135

Kaufmann G, Vosburg SK (1997) Paradoxial mood effects on creative problem solving. Cogn Emotion 11(2):151–170

Khan IA, Brinkman WP, Hierons RM (2007) Moods and programmers' performance. 22nd Psychology of programming conference, Joensuu, Finland, pp 3–16

Khan IA, Brinkman WP, Hierons RM (2008) Toward a mood measuring instrument from interaction behavior. In: 23rd conference on psychology of programming, Lancaster University, Lancaster

Kirchsteiger G, Rigotti L, Rustichini A (2006) Your morals might be your moods. J Econ Behav Organ 59:155–172

Kleinginna PR Jr, Kleinginna AM (1981) A categorized list of emotion definitions, with suggestions for a consensual definition. Motiv Emotion 5(4):345–379

Krantz JH, Dalal R (2000) Validity of web-based psychological research. In: Birnbaum MH (ed) Psychological experiments on the internet. Academic Press, USA, pp 35–60

Lafore R (1998) Data structures and algorithms in java. Waite Group Press, Corte Madera

Lane AM, Whyte GP, Terry PC, Nevill AM (2005) Mood, self-set goals and examination performance: the moderating effect of depressed mood. Person Individ Differ 39(1):143–153

Lang PJ (1980) Behavioral treatment and bio-behavioral assessment computer applications. In: Sidowski JB, Johnson JH, Williams TA (eds) Technology in mental health care delivery

Lang PJ (1995) The emotion probe: studies of motivation and attention. Am Psychol 50(5):372–385

Larson GE, Alderton DL, Neideffer M, Underhill E (1997) Further evidence on dimensionality and correlates of the Cognitive Failures Questionnaire. Br J Psychol 88(1):29–38

Lewis PA, Critchley HD (2003) Mood dependent memory. Trends Cogn Sci 7:431–433

Lowther J, Lane A (2006) Relationship between moods. Cohesion and satisfaction with performance among soccer players. Athletic Insight Online J Sport Psychol 4(3):18

Lui KM, Chan KCC (2004) A cognitive model for solo programming and pair programming. In: Proceedings of the third IEEE international conference on cognitive informatics, (ICCI'04), pp 94–102

MacWhinney B, Keenan JM, Reinke P (1982) The role of arousal in memory for conversation. Cogn Emotion 10(4):308–317

Martin LL (2001) Mood as input: a configural view of mood effects. In: Martine LL, Clore GL (eds) Theories of mood and cognition. Lawrence Erlbaum, Mahwah, pp 135–158

Mayrhauser AV (1995) Program comprehension during software maintenance and evolution. IEEE Comput 28(8):44–55

McMorris T, Graydon J (1997) The effect of exercise on cognitive performance in soccer specific tests. J Sports Sci 15(5):459–468

Morris JD (1995) SAM: The self-assessment manikin, an efficient cross-cultural measurement of emotional response (observations). Journal of Advertising Research 35(6):63–68

Mujica-Parodi L, Greenberg T, Kilpatrick J (2004) A multi-modal study of cognitive processing under negative emotional arousal. In: 26th annual meeting of cognitive science society

Nancy LL, Barrett KC, Morgan GA (2005) SPSS for intermediate statistics: use and interpretation. Lawrence Erlbaum Associates, Mahwah, NJ

Nasoz F, Alvarez K, Lisetti CL, Finkelstein N (2003) Emotion recognition from physiological signals for presence technologies. Int J Cogn Technol Work (special issue on presence) 6(1):1–32

Norman DA (2004) Emotional design: why we love (or hate) everyday things. Basic Books, New York

Parberry I, Gasarch W (2002) Problems on algorithms. Prentice Hall Inc, Englewood Cliffs

Parkinson B, Totterdell P, Briner RB, Reynolds S (1996) Changing moods: the psychology of mood and mood regulation. Longman, Harlow

Pearnkandola (2007) Mood and emotions at work. Retrieved October 2009 from pearnkandola website http://www.pearnkandola.com/research-panel/4.html

Plutchik R (1980) A psychoevolutionary theory of emotions. Social Sci Inf 21(4/5):529–553

Reason JT (1995) Human error. Cambridge University Press, Cambridge

Reips UD (2000) The web experiment method: advantages, disadvantages, and solutions. In: Birnbaum MH (ed) Psychological experiments on the internet. Academic Press, San Diego, pp 89–117

Remington NA, Fabrigar LR (1995) Re-examining the circumplex model of affect. J Pers Soc Psychol 79(2):286–300

Robelin M, Rogers PJ (1998) Mood and psychomotor performance effects of the first but not of subsequent cup-of-coffee equivalent doses of caffeine consumed after overnight caffeine abstinence. Behav Pharmacol 9(7):611–618

Rocco E, Igou B (2001) The critical role of high-availability infrastructure support services. Gartner Research, Stamford

Roets A, Hiel AV (2008) Why some hate to Dilly-Dally and others do not: the arousal-invoking capacity of decision-making for low- and high-scoring need for closure individuals. Social Cogn 26(3):333–346

Rugaber S, Tisdale VG (1992) Software psychology requirements for software maintenance activities. Software Engineering Research Center, Georgia Institute of Technology

Russ SW, Kaugars AS (2001) Emotion in children's play and creative problem solving. Creativ Res J 13:211–219

Russell JA, Barrett LF (1999) Core affect, prototypical emotional episodes, and other things called emotion: dissecting the elephant. J Pers Soc Psychol 76:805–819

Rusting CL (1998) Personality, mood, and cognitive processing of emotional information: three conceptual frameworks. Psychol Bull 124:165–196

Sanchez JA, Kirschning I, Palacio JC, Ostrovskaya Y (2005) Toward mood-oriented interfaces for synchronous interaction. In: Proceedings of 2005 Latin American conference on human-computer interaction CLIHC'05, Cuernavaca- Mexico, 124, pp 1–7

Schmitt R (1969) Martin Heidegger on being human: An introduction to "Sein and Zeit". Random House, New York

Schneiderman B, Mayer JE (1979) Syntactic/semantic interaction in programmer behaviour: A model and experimental results. Int J Comput Inf Sci 8(3):219–238

Shaochun X, Rajlich V (2004) Cognitive process during program debugging. In: Proceedings of the third IEEE international conference on cognitive informatics, 176−182

Shaw T (2004) The emotions of systems developers: an empirical study of affective events theory. In: Proceedings of 2004 SIGMIS conference on computer personnel research, 124–126

Shepard RJ (1997) Curricular physical activity and academic performance. Paediatr Exercise Sci 9(2):113–126

Silvestrini N, Gendolla GHE (2007) Mood effects on autonomic activity in mood regulation. Psychophysiology 44:650–659

Smith HJ, Keil M (2003) The reluctance to report bad news on troubled software projects: a theoretical model. Inf Syst J 13:69–95

Smith AP, Brockman P, Flynn R, Maben A et al (1993) Investigation of the effects of coffee on alertness and performance during the day and night. Neuropsychobiology 27(4):217–223

Steptoe A, Cox S (1988) Acute effects of aerobic exercise on mood. Health Psychol 7(4):329–340

Thayer RE, Newman JR, McClain TM (1994) Self regulation of mood: strategies for changing a bad mood, raising energy, and reducing tension. J Pers Soc Psychol 67(5):910–925

Vaa T (2007) Modelling driver behaviour on basis of emotions and feelings: intelligent transport systems and behavioural adaptations. In Cacciabue PC (ed) Modelling driver behaviour in automotive systems—critical issues in driver interactions with intelligent transport systems. Springer, London 2007, pp 208–232

Watters PA, Martin F, Schreter Z (1997) Caffeine and cognitive performance: the nonlinear Yerkes-Dodson Law. Human Psychopharmacol 12:249–257

Weiss RP (2000) Emotion and Learning—implications of new neurological research for training techniques. Training and Development

Weiss HM, Cropanzano R (1996) Affective events theory: a theoretical discussion of the structure, causes, and consequences of affective experiences at work. Res Organ Behav 18:1–74

Yerkes RM, Dodson JD (1908) The relation of strength of stimulus to rapidity of Habit-Formation. J Comp Neurol Psychol 18: 459–482

Zheng W (2008) The Effects of Message Arousal and valence on Implicit and Explicit Memory of Pass. In: Annual meeting of the international communication association, Dresden, Germany, Retrieved October 2009 from allacademic.com website url: http://www.allacademic.com/meta/p91350_index.html

Zimmermann P, Guttormsen S, Danuser B, Gomez P (2003) Affective computing—a rationale for measuring mood with mouse and keyboard. Int J Occup Safe Ergonomics 9(4):539–551