

Free University of Bozen-Bolzano
Faculty of Computer Science



Thesis

The Impact of Affective States on Skills
and Productivity in Software
Development

Daniel Graziotin, B.Sc.
Thesis Advisor: Prof. Pekka Abrahamsson
July 2012

Abstract

There is a constant need to improve the way the software is being developed due to growing needs of software in everyday life and devices. Even if there is an emerging, yet consolidated desire to consider software developers as individuals, software engineering (SE) research forgets that they are not industrial, unit-of-work machines. Software Developers are human beings. They solve problems in creative ways. For 20 years, human-related productivity factors in software development have been gaining attention. However, there are no factors that take into account the diversity of developers and their unique, complex minds.

There is a well-established body of knowledge in Psychology and Cognitive Science that studies how the affective states of the individuals are related to their performance at work. Unfortunately, there are no studies to merge and test these theories in SE. In particular, the definition of the productivity of software developers is an open problem for more than 30 years. The mood of software developers is completely ignored.

Affective states (feelings, emotions, and moods) are important. There are numerous studies not related to SE claiming that affective states have an impact on creativity, on the capacity of solving problems and on productivity. We claim that affective states can not be ignored by SE. Additionally, there is a recent call to adopt psychometrics in Empirical Software Engineering studies, to address the uniqueness of each individual participating in a Software Development Life-cycle. The discipline is tentatively called Individualized Software Engineering.

In this study we pose a question that must be answered by SE via many present and future researches. The question is “What is the role of affective states for software developers?”. To provide first insights for answering the question, we conducted two empirical studies. The first study is related to how affective states are indicators of creativity and problem solving skills of software developers. In the second study we propose a linear mixed effects models that estimates the productivity of software developers by measuring their affective states related to the task they perform. We test the model with a repeated measures, longitudinal experiment in order to provide empirical results.

We found significant evidence that software developers feeling positive affective states also possess higher problem solving skills. We believe that the same holds for the creativity, but we could not find significant evidence. Nevertheless, the analysis of the data encourages future research. We found significant, positive correlation with the affective states of software developers and their task productivity. We call for repetition of the experiment to further investigate our claims.

The results of our studies are relevant for SE researchers, for Computer Science Education researchers and lecturers, as well for the Information Technology Industry. Information Technology firms may test the long-term affective states in the hiring phases of potential workers to predict their skills. Future studies in SE should research about how to raise positive affective states of developers and assess their skills and productivity. Future research may also lead to the creation of a feedback-based system for IT companies to assesses the productivity at different organizational-levels by measuring the mood of workers.

Acknowledgments

This thesis required more than one year of work for me. So many people made this work possible and it is a pleasure to thank all of them. I am very grateful to everybody who contributed to my research, even those that have been forgotten here because of my distraction.

It is difficult to find the words to express my gratitude to my M.Sc. supervisor, prof. Pekka Abrahamsson. Besides all the gifts, the qualities and the effort you would expect to find in a thesis supervisor, Pekka Abrahamsson showed great enthusiasm, inspiration, and support for my research interests. Throughout my thesis-writing period, he provided encouragement, sound advice, many powerful ideas, and good company. So many things would be missing without his presence. Thank you, Pekka.

The S4FS research team at the Faculty of Computer Science of the Free University of Bozen-Bolzano was like a second home for me, for a year long. All the researchers in there provided me many useful advices and always made me feel like a regular researcher and colleague. I am very grateful to Xiaofeng Wang, for so many reasons. Among them, I am grateful for her support in setting up the first experiment of the thesis (and of my whole life). Then, I would like to thank her for the huge effort in setting up the research hypotheses on the thesis and the correction of more than 80 pages, word by word. You acted like a co-supervisor for me, thank you very much. I would like to thank Juha Rikkilae for setting up the first experiment as well, and for being an exceptional desk neighbor. I would like to acknowledge Cigdem Gencel for her presence and the initial discussions to find the perfect regression model for the second study. Thank you Nattakarn “Joy” Phaphoom for being the other exceptional desk neighbor and for making me discover the work of Mihaly Csikszentmihalyi.

Other researchers and students of the Free University of Bozen-Bolzano deserve my gratitude. Among them, I would like to thank Ilenia Fronza for her support with Statistics and for being a terrific companion for coffee breaks. Also, I would like to thank my friend Roberto Gritti for his constant presence before, during, and after this experience.

I am grateful to all the people of the Faculty Staffs (Administration of the Faculty and the Computer Pool of the Faculty), for their effort in the organization of the first experiment and for their very good company. Among them, I would like to particularly thank Federica Cumer and Cristiano Cumer.

I would like to thank all the participants of both experiments, as well as the six judges experts in the field of creativity. Without your presence, this thesis would not exist. Among them, my special acknowledgments are for prof. Matteo Moretti of the Faculty of Design and Arts. Besides being an exceptional workmate, Matteo helped me to contact many other judges and to setup the creativity task of the first experiment.

I am grateful to researchers at the Fondazione Bruno Kessler, as well. My

former B.Sc. supervisor Paolo Massa and my former tutor Maurizio Napolitano played an important role for networking with experts in the research fields related to this thesis and provided interesting inputs and advices. I would also like to thank Kyriaki Kalimeri, for her advices about affective states measurements.

It is time for me to thank people who I hold in my heart. I could never express my gratitude to them with English, so I will switch to Italian.

Vorrei ringraziare Matteo Dell’Osso per l’ospitalità e la compagnia durante il primo anno di Master, oltre che per le uscite serali in compagnia. Senza contare la sua abilità, disponibilità e simpatia come insegnante di basso elettrico.

Voglio ringraziare di cuore la mia famiglia per l’appoggio, il supporto (e la sopportazione!) che durano ormai da 25 anni. Mi hanno incoraggiato a seguire le mie aspirazioni per tutta la vita ed è soprattutto grazie a loro se sono arrivato a questo punto.

Infine, ma non di certo per ordine di importanza, voglio ringraziare di cuore Elena. Per ogni singolo istante passato insieme, per l’appoggio e per avermi donato una gioia di vita inestimabile.

È a lei che dedico questa tesi.

Contents

1	Introduction	10
1.1	Motivation	10
1.2	Scope of the Research	12
1.3	Research Questions	12
1.4	Structure of the Thesis	13
2	Related Work and Background Research	15
2.1	Software Engineering studies	15
2.1.1	Productivity Formulas	15
2.1.2	Productivity Factors	17
2.1.3	The Human Aspect of Software Developers	18
2.2	Psychology and Cognitive Science studies	20
2.2.1	Affective States, Emotions, Moods	20
2.2.2	Affective States in Creativity, Problem Solving Skills, and Productivity	20
2.2.3	Measuring Affective States, Creativity, and Problem Solving Skills	22
2.2.4	Affective States and Software Development	22
3	Research Framework	24
3.1	Theoretical Model	24
3.2	Research Hypotheses	26
4	Empirical Research Design	28
5	Empirical Results	29
5.1	Study 1: Affective States and Software Development Skills	29
5.1.1	Introduction	30
5.1.1.1	Motivation	30
5.1.2	Related Work	31
5.1.3	Description of the Experiment	31
5.1.3.1	Goal Definition	31
5.1.4	Experimental Design	32
5.1.4.1	Hypothesis Formulation	33
5.1.4.2	Variables	34
5.1.4.3	Participants Units	35
5.1.4.4	Objects and Instrumentation	35
5.1.4.5	Validity Evaluation	35
5.1.4.6	Execution	36
5.1.5	Analysis	36

5.1.5.1	Descriptive Statistics	37
5.1.5.2	Data Reduction	46
5.1.5.3	Hypothesis Testing	46
5.1.6	Discussion	48
5.1.6.1	Limitations of the Study	49
5.1.6.2	Lessons Learned	50
5.1.7	Primary Empirical Conclusions	51
5.2	Study 2: Affective States and Software Development Productivity	52
5.2.1	Introduction	53
5.2.1.1	Motivation	53
5.2.2	Related Work	54
5.2.3	Description of the Experiment	54
5.2.3.1	Goal Definition	55
5.2.3.2	Experimental Design	56
5.2.3.3	Hypothesis Formulation	56
5.2.3.4	Variables	56
5.2.3.5	Linear, Mixed Effects Models	57
5.2.3.6	Participants Units	57
5.2.3.7	Objects and Instrumentation	58
5.2.3.8	Validity Evaluation	58
5.2.3.9	Execution	58
5.2.4	Analysis	58
5.2.4.1	Descriptive Statistics	58
5.2.4.2	Hypothesis Testing & Model Estimation	63
5.2.5	Discussion	65
5.2.5.1	Limitations of the Study	66
5.2.6	Primary Empirical Conclusions	66
6	Discussion	68
6.1	Theoretical and Practical Implications	70
7	Conclusions	70
7.1	Answers to the Research Questions	71
7.2	Future Research Opportunities	72
	References	74
	Appendix	79
A	Experiment Instructions for Participants	79
B	Experiment Instructions for Creativity Judges	81
C	Experiment Instructions for the Problem-Solving Task	81
D	Scale of Positive and Negative Experience (SPANE)	81

E	Experiment Instructions for the Mood Dimensions vs. Productivity Study	85
F	Mood Dimensions vs. Productivity Study Survey	87

List of Figures

1	Model for Study 1	25
2	Model for Study 2	26
3	Tower of London Game	37
4	Affect Balance Score for the Creativity Task	38
5	Box-plots for the number of generated ideas of N-POS and POS groups	39
6	Box-plots of the BCreat score for the N-POS and POS groups . .	40
7	Scatter-plot BCreat score vs. SPANE-B	41
8	Scatter-plot μ_{Creat} for the N-POS and POS groups	42
9	Scatter-plot μ_{Creat} vs. SPANE-B for the N-POS and POS groups	43
10	Affect Balance Score for the Problem-Solving Task	44
11	Box-plots of the ProbSolv score	45
12	Scatter-plot Creat vs. SPANE-B	46
13	Productivity & Valence over Time	61
14	Productivity & Arousal over Time	62
15	Productivity & Dominance over Time	63
16	Productivity vs. Mood Dimensions Survey	87

List of Tables

1	t-test for $\mu_{BCreat_{N-POS}}$ and $\mu_{BCreat_{POS}}$	47
2	t-test for $\mu_{Creat_{N-POS}}$ and $\mu_{Creat_{POS}}$	47
3	t-test for $\mu_{num.captions_{N-POS}}$ and $\mu_{num.captions_{POS}}$	47
4	t-test for $ProbSolv_{N-POS}$ and $ProbSolv_{POS}$	48
5	Participants	59
6	Projects and Tasks	59
7	Productivity, valence, arousal, and dominance mean, std. dev., and median	60
8	Full-Model Coefficient Estimation, *p < 0.05	65

1 Introduction

In this section, we outline the motivation of our work, the research questions, and the scope of the research of this thesis.

1.1 Motivation

There is a constant need to improve the way the software is being developed due to growing needs of software in everyday life and devices. One of the central roles of software engineering is to define and measure the improvements of the software product and the development process. In particular, the productivity (or performance) of software developers is a controversial debate since more than 40 years.

The measurement of the work accomplished by developers is still an open problem. The simplest, classic definition of programmer’s productivity is a ratio-based formula, defined in terms of the number of lines of code (LOC) divided by a unit period of time (Chen, 1978). Unfortunately, the number of lines of code does not represent real effort of a programmer. The effort needed when developing software depends on the problem being solved and will vary with the complexity of each task (Collofello and Woodfield, 1983). Additionally, two developers will implement different solutions for the same problem, using different numbers of LOC and providing different levels of efficiency.

There are many attempts to substitute the number of source code statements with other measurements like functions points or Software Science, but they are unsuccessful (Collofello and Woodfield, 1983). The modern proposals range from weighting productivity factors to Data Envelopment Analysis, Bayesian Networks and Statistical Process Control. All the proposed models either have limitations or need further research (Petersen, 2011).

What all the proposed models fail to take into consideration is that the work is accomplished by humans, all of which are different and unique. The issues related to define a universal, comparable software productivity model led to the proposal of higher-level views to focus on the utility of delivered product, rather than on the amount of software written in order to deliver the product (Dale and Van Der Zee, 1992).

While there is a long debate about productivity models, there is agreement about the existence of several factors that influence the productivity of developers. More than 50 factors influencing productivity emerge from recent literature reviews (Sampaio et al., 2010, Wagner and Ruhe, 2008).

There is an emerging, yet consolidated desire to consider software developers as individuals (Beck et al., 2001). Agile Software development relies on people, their talents and their skills, rather than the process itself (Cockburn and Highsmith, 2001). Indeed, software developers are human beings. They solve problems in creative ways. Only recently, human-related productivity factors such as motivation, quality of management (Sampaio et al., 2010), friendly atmosphere, support for innovation, and creativity (Wagner and Ruhe, 2008) are gaining attention.

Programmers transform general - sometimes informal - understanding of a goal into a formal model that is understandable by the computer (Fischer, 1987). This process of transformation is the result of activities carried on through skills of cognitive processing and creativity.

We are all able to rate our productivity in a general way. We all have a feeling that our software development today is “on the average”, like we expect it to be. We can state that our performance is “worse than usual”, or “incredibly high”. There must be a reason behind. For the purposes of this investigation we do not differentiate the terms affective states, emotions, moods, and feelings, but we provide additional details and definitions in section 2.2.1.

We believe that affective states play an important role in the productivity of software developers, their problem-solving skills and their creativity.

Since the beginning of the past century, Psychology and Cognitive Science study the correlation between emotions of individuals and their work-related performance, their cognitive processing¹, and their creativity skills. Affective states of the individuals have influence on work-related behaviors and capacities (Ilies and Judge, 2002).

There is no strong agreement in Psychology about the impact that the affective states have on the creativity of individuals. Many studies find a positive correlation between the emotions and the creativity (Baas et al., 2008, Davis, 2009). Other studies find a positive correlation between negative affective states and creativity, as well.

Analytical problem-solving capacities - thus, not related to creative problem solving - are believed to be higher when feeling negative emotions (Abele-Brehm, 1992, Melton, 1995). However, the process of learning and cognitive processing are lowered by negative emotions (Brand et al., 2007).

Software engineering lacks synergies with other disciplines in order to further investigate about creativity, problem-solving, and productivity. Problem-solving capacities and creativity are important for software development. There are few studies about the role of affective states in software development tasks (Khan et al., 2010, Shaw, 2004). There are no studies about the impact of emotions on skills and productivity of software developers. But something new is happening.

There is a recent call to study psychometrics of the individuals in future software engineering empirical studies.

“Software Engineering researchers should put a larger focus on the humans involved in software development than what has been done to date. One easy and powerful way to do this would be to collect psychometric measurements”. (Feldt et al., 2008)

The call refers to the creation of a branch of software engineering, tentatively called Individualized Software Engineering (Feldt et al., 2008). We accept the

¹From this point on, we will refer to cognitive processing, analytical problem solving and planning capacities with the term problem-solving.

challenge and provide new evidences that psychometric research in the field of software engineering is important and relevant.

The aim of this thesis is to lay down the basic building blocks on the impact of affective states for software developers as individuals.

In the following sub-sections, we state the research questions of this thesis, the scope of the study, and the structure of this document.

1.2 Scope of the Research

Our work contains two empirical studies. The first study is a research on the role of long-term affective states in the skills of problem-solving and creativity of software developers. It is a controlled experiment. The second work is a study on the correlation of the real-time mood of software developers about their task and their immediate productivity. The experiment is a within-subjects, repeated measurements design. Additionally, the study provides empirical evidences of the use of a linear mixed effects model to estimate the productivity of software developers using three mood dimensions as predictor variables.

The studies concentrate on affective states, creativity, problem solving, and productivity at the individual levels. The study on creativity and problem solving evaluates these two skills by assigning a score to two tasks faced by each participant. The two scores represent general measures of creativity and analytical problem-solving. The study on the correlation between mood and real-time productivity analyzes individual developers (professional and students) while they work on their personal or professional projects. The study is solely related to the programming activity. Moreover, it is not our scope to compare affective states with the already existing software productivity factors.

It is not a scope of this work to analyze teams of developers. All the studies are aimed to analyze developers as single individuals. Additionally, it is not a scope of this thesis to study causalities. This thesis studies correlations of affective states with skills for software development and productivity while programming.

We provide further details in the *Limitations* sections of the two studies.

1.3 Research Questions

In this thesis we investigate about the role of emotions in Software Development. We lay ground the research to provide answers to the question:

“HOW DO THE AFFECTIVE STATES IMPACT ON SKILLS AND
PRODUCTIVITY IN SOFTWARE DEVELOPMENT?”

The research question will be answered by the following three sub questions:

1. How do pre-existing affective states influence the creativity of software developers?

2. How do pre-existing affective states influence the analytical problem-solving skill of software developers?
3. How do real-time affective states influence the immediate productivity of software developers?

1.4 Structure of the Thesis

In this sub-section we explain the structure of the study. We reported the motivation, the research questions, and the scope of the research in section 1.

Section 2, called Related Work and Background Research, contains our review of the background research on which we rely to create the research framework. It is divided in two areas. The first area is related to productivity in software engineering and the recent desire to treat software developers as human beings. The second area is dedicated to the research in Psychology and Cognitive Science about affective states (emotions, mood) and their relationship with creativity, problem solving, and productivity.

The Research Framework (section 3) contains a conceptualization of our understanding of the related work and how we infer the research questions. Moreover, we explain the reasons that brought us to divide the research in two different empirical methods.

In section 4, we provide the empirical strategies available to software engineering and which one we decided to adopt for our two experiments. Additionally, we provide the theoretical background behind linear mixed models for repeated measurements, longitudinal, within-subjects designs.

Section 5 (Empirical Results) provides our two experiments. They are reported as almost self-contained research papers. We wrote the papers following the guidelines of Wohlin et al. (2000) and Jedlitschka et al. (2008). Both studies begin with a structured abstract, an introduction to the problem researched and the motivation. What follows is the Related Work section, in which similar studies are analyzed and reported. In the Experimental Design section, we report the outcomes of the planning phase of the experiment. Therefore, we write the Research Goals, the Hypothesis, the Variables, and the Experiment Design. Then, we inform the reader about the participants of the experiment, objects and instrumentation required, the analysis procedure, the validity evaluation, and a description of the tasks to be performed. In the Analysis section, we describe all the analysis performed against the gathered data. First, we provide the descriptive statistics. Then, we report the statistical tests against the hypotheses. What follows the Analysis section is the Discussion section, where we discuss the results obtained from the data and the hypothesis testing. Additionally, we compare our results with the related work. The final section is the Primary Empirical Conclusions (PEC) section, in which we briefly report which are the implications of our findings. We keep them short, because they will be recalled in the Conclusions section of the thesis. To avoid additional redundancy with the thesis, we also reduce the sections Introduction and Related Work of

the two papers. In these sections we report the relevant facts that are already mentioned in Sections 1 and 2 of the thesis and the results of the related works.

Section 6 (Discussion) expands the PEC of the two papers. We also state the theoretical and practical implications of our work.

Section 7 contains the Conclusions of the thesis. In this section we summarize the thesis and we provide our answers to the research questions. Additionally, we present the future research opportunities.

After the Conclusions of the thesis, we provide the bibliography of the referenced studies and the Appendix.

As additional note, the reader may notice the adoption of unusual colors for our graphs and models. We followed guidelines to let color-blind people notice the difference between elements of the figures (Okabe and Ito, 2008).

2 Related Work and Background Research

In this section we provide the background of our research and the related work that lead to the creation of our theoretical framework. This section is divided in two areas. The first area is related to the body of knowledge regarding productivity in software engineering (2.1). The second area is about affective states, software development skills, and productivity in Psychology and Cognitive Science (2.2).

In sub-section 2.1, we summarize the state of the art of productivity formulas and productivity factors in software engineering. Then, we provide the studies on the need to treat software developers as human beings.

In sub-section 2.2, we report our knowledge about affective states and we provide definitions for emotions and moods. Thereafter, we summarize the state of the art on the role of affective states, creativity, problem-solving, and productivity. The following sub-section is to report the measurement instruments for affective states, creativity, and problem-solving. The last sub-section is about the research in Psychology and Cognitive Science about software development.

2.1 Software Engineering studies

In this sub-section, we report the body of knowledge related to the research in software development productivity. In the first part, we describe the attempts to measure the productivity of software developers. Then, we provide a summary of the research behind the factors that influence productivity when developing software. In the last part, we provide our observations about the ever growing desire to study the human aspects of software developers.

2.1.1 Productivity Formulas

The expansion of software firms has significantly expanded since the 70's. Nowadays, software industries are propagated everywhere and essential for all the other industry fields. With all the software products released every year in the world, there is interest in productivity of software developers. A productivity measure is understood as a ratio of outputs produced to resources consumed. The ability to measure the productivity in software development may be used in many different ways, like project estimation and process evaluation. Therefore, to have a definition for the productivity of software developers is crucial for decision makers.

A formula to determine productivity and to compare the productivity at different levels of a firm organization and between firms is still an open problem for software engineering.

The first definition of productivity in software followed the economics model of effort over time. Productivity was expressed as valid source statements written by the programmer during a unit period of time (Chen, 1978). This simple, classic formula, $Productivity = \frac{SizeInLoc}{EffortInTime}$, faces many issues. For example, the effort needed when developing software depends on the problem being

solved and will vary with the complexity of each task (Collofello and Woodfield, 1983). Additionally, two developers will implement different solutions for the same problem, using different numbers of LOC and providing different levels of efficiency. Finally, the number of lines of code can only be determined with confidence near the completion of the project.

The numerous issues behind the classic productivity formula were immediately understood. In fact, the research specialized in two different aspects of productivity. Many studies attempt to either enhance the classic formula or to introduce new models. Other research identifies the factors that influence the productivity of developers.

The enhancements of the classic formula comprise new variables such as the size of the problem and the quality of the output (Chrysler, 1978), where the problem size is determined at the requirements level but the quality is yet another an open problem in software engineering. Other attempts to determine the effort needed are the use of function points (Albrecht, 1979) or software science (Halstead, 1977). Although a correlation between these measures and the number of source statements is established (Albrecht and Gaffney, 1983), the function point formula incorporates weights but we do not know how these weights were obtained. Additionally, function points do not consider the programmers as human beings and do not take into account the factors that influence productivity. There are other numerous studies, but none of them affirmed their proposal to measure the productivity. In 1992, a standard proposal for software productivity metrics attempted to end the need for research with some factors and variations, such as delivered code, reused code, etc.. It provides a form with project characteristics, but does not discuss how different characteristics influence the productivity. The standard has been reaffirmed in year 2002 and withdrawn in year 2008 (IEEE, 1992).

The attempts to overcome the limitation of the ratio-based formula are more recent but variegated. There are proposals to analyze the earned value as the percentage of progress towards the final product (Kadary, 1992). Some studies introduce the use of non-parametric analysis like Data envelopment analysis, where inputs and outputs are weighted and their ratio is maximized (Ding et al., 2006). There are proposals to use Bayesian Belief networks, where productivity is represented as a directed graph in which the nodes are productivity-related variables connected by cause-effect relationships (Stamelos et al., 2003). There are numerous other proposals. All of the before mentioned studies are promising but either lack in their validity threats evaluations or have insufficient number of empirical studies behind (Petersen, 2011). Additionally, the studies of the productivity factors highlight too many variables (Scacchi, 1995). The perfect productivity model does not yet exist.

To summarize, we have many proposals to define and measure software development productivity. Unfortunately, we still do not know what software productivity is. All the proposed measurements, even if promising, are missing empirical foundations and validity. Additionally, they lack in the ability to generalize the values and to let the decision makers perform comparisons among firms.

2.1.2 Productivity Factors

The studies on the factors influencing software development productivity started at about the same time of the first attempts to define measurements for productivity. Already more than 30 years ago, programmer's productivity was believed to be influenced by different characteristics. The characteristics are at the technical level (programming mode, programming language, hardware), at the knowledge level (domain of interest, organizational operations), and at the programmer level (only skills and experience measured in number of months) (Chrysler, 1978). The study performs an empirical research in order to determine the effect of various factors that influence productivity. The results report that the most valuable factors are programmer's experience at the current facility, the number of input files, the number of control breaks and totals, the number of input edits, and the number of input fields. But the programmers were not required to log their activities while working. They only had submit the number of hours on a weekly basis. Additionally, just the number of years of education of the programmer are taken in consideration, not the quality of the education received.

In the mid-80's, another study criticizes the previous research. The previous studies are very weak in terms of measures and data observed (Collofello and Woodfield, 1983). But in the analysis, it concludes that no single technology guarantees productivity gains and that large money investments must be done in order to improve productivity.

During the 90's the work of Barry Boehm (1990) identifies other factors influencing the productivity by changing the perspective: to control the costs of software. Together with other previous studies and the COCOMO model, the study indicates strategies to improve the productivity such as writing less code and getting the best from people (Boehm, 1990). We will re-consider this article in the next section.

The work of Scacchi (1995) is a study about what affects software productivity and how it can be improved. It is the first noticeable literature review about productivity factors. It focuses on the creation of a framework to predict the productivity of large-scale software systems. The study criticizes the previous studies, because they fail to describe the variation in productivity among individual programmers. Moreover, the number of involved variables and factors implies that the productivity can not be measured as a ratio of lines of code per time. Regarding the software productivity drivers, an important impact is attributed to human-related factors. Organizational and social conditions can even dominate the productivity contribution attributable to in-place software development technologies. The study calls for alternative directions for software productivity measurements and improvement (Scacchi, 1995).

A review of productivity factors by Sampaio et al. (2010) identifies three main areas in the body of knowledge: product, project, and people. The product related factors comprise reuse of software components, the size of the product, and the complexity of the program. Project factors are composed by the team size, the stability of the requirements, the experience and the participation of the

customer to the development life-cycle, the process complexity, the tools, and the programming languages. The people related factors consist in the motivation of the team, the building of the team in terms of skills but also relationships, and the quality of management. The study complains about lacks in quantifying the impacts and calls for the creation of a universal catalog of productivity factors (Sampaio et al., 2010).

We see that there is an increasing awareness of the *human* factor in software development. The next section deals with the studies that are built around the concept that software developers are human beings and as human beings they have specific qualities and needs.

2.1.3 The Human Aspect of Software Developers

As we anticipated in sub-section 1.1, there is an emerging desire to consider software developers as unique, different humans rather than industrial-like machines. The Agile Manifesto (Beck et al., 2001) was published because of the value of “Individuals and interactions over processes and tools”. Before the arrival of Agile, traditional software development processes considered the developers as tools to achieve the deliverables, being them requirements, design documents, test cases, the final program, etc..

The Agile movement and the successive Agile software methodologies such as Scrum (Schwaber and Beedle, 2001) and eXtreme Programming (Beck, 1999) put the human being as the main role, around which everything else is built. For example, Scrum is a project management tool in which the communication among people is essential at different levels (Daily Scrum meeting, Scrum of Scrums, Sprint planning/review meeting, ..). The eXtreme Programming methodology was created as a response to changing requirements which are mainly caused by the human nature of the customer and the people in the whole environment. It proposes pair programming, where two developers must communicate and perform extensive code review, develop and create unit testing of all the code. In fact, the value of the individuals and interactions over process cause the Agile proposals to be rather called methodologies instead of processes.

An article of Cockburn & Highsmith (2001) is entitled “Agile Software Development: The People Factor”. It suggests to reduce the cost of moving information between people by making them physically closer, to replace documents with meetings, and improve the sense of the team as a community. Agile development focuses on the talents, the skills, and the individual competence as a critical factor in the project success. The importance given to the individuals is higher than the process itself. The entire Agile ecosystem is built from people with different personalities and different skills: everybody is important and everybody is part of the ecosystem itself (Cockburn and Highsmith, 2001).

As we anticipated in the previous section, Boehm (1990) indicates strategies to improve the productivity. Many of these strategies are people-centered. The author states that many software programs will not be developed because of time-constraints. However, raising the efficiency of good software developers by

enhancing their way of working will provide time for them to fulfill the demand for software. Then, the productivity when developing software and the quality of the produced software largely vary according to the differences between people. The secret to achieve low cost but high quality software is to write less code and to make people more effective. *Writing less code* implies to build simpler products and re-use software components. *Making people more effective* is one of the first signals in treating developers as human beings. In fact, some of the provided suggestions are to assign developers private offices, to create a working environment to support creativity, and to provide incentives (Boehm, 1990).

Software Process Improvement (SPI) research began in the last decade to consider the commitment of people a necessary factor to determine the success of well-planned process improvement programs. Commitment is commonly considered as a psychological state of attachment between a person and an entity, where an entity can be a person, an organization, a team, or the project itself. Commitment may take different forms but among them, the most close to our study is the affective component. Affective commitment refers to the employee's attachment to, identification with, and involvement within the entity in question. There is a need to study the commitment of the individuals involved in SPI initiatives, as commitment determines success (Abrahamsson, 2001).

The Agile movement is another improvement to rely on individuals rather than industrial machines. The desire of software developers to be treated as human beings is so strong that a recent movement was born, called "Programming, Motherfucker"². The Manifesto is a criticism of the Agile movement, because it accuses the Agile methodologies to fail in all their proposed human-related values .

We can find an analysis of the individual characteristics in the work of Darcy (2005). It researches on the impact of personalities on programming performance. In this study, performance is defined as the degree to which a software adheres to the specification and the complexity of the software itself. The work presents a quasi-experiment in which 29 students in Computer Science were asked to implement a solution to the same problem, using the same tools and the same programming language (C++). Among them, 5 students could complete the task and 24 submitted non-compilable or incomplete solutions. The analysis suggested that there is not evidence for a correlation with the personalities of the individuals and the final results. However, only a tiny number of students completed the task and the author found homogeneity in the personality types. Nevertheless, the study calls for future research about individual characteristics and software development tasks (Darcy et al., 2005).

The work of Shaw (2004) is a "research-in-progress" paper reporting the need to study the emotion of system developers. The paper does not report about empirical evidences to answer specific research questions. Nevertheless, it observes that in 48 hours of interval, there are drastic changes in the mood of software developers (Shaw, 2004).

²Shaw, Programming, Motherfucker - <http://programming-motherfucker.com/> (last accessed July, 2012)

2.2 Psychology and Cognitive Science studies

Psychology and Cognitive Science study individual performance and skills with respect of many psychometrics. Affective states are among them. This subsection contains our body of knowledge research about affective states, the role of affective states for important skills in software development and productivity, and a review of the most important measurement tools for affective states and skills.

2.2.1 Affective States, Emotions, Moods

While psychometrics are temporarily ignored by software engineering, Psychology and Cognitive Science have got a long story of studies in the field. In particular, the topic cognition refers to an information processing view of an individual's psychological functions. It is difficult to differentiate terms like affective state, emotion, and mood. Emotions are states of mind that are raised by external stimuli and are directed toward the stimulus in the environment by which they are raised (Plutchik and Kellerman, 1980). Moods are affective states in which the individual feels good or bad, and either likes or dislikes what is happening around him/her. Moods are believed to last longer than emotions (Parkinson et al., 1996). There is no clear agreement on a differentiation of the terms emotion and mood. Many authors consider mood and emotions as the same entity (Dow, 1992). Therefore, in this thesis we use the terms affective state, mood, and emotion interchangeably. Because the literature reports about emotions (moods) in the long term and in the short term, we coherently differentiate in terms of time.

There are two main theories to categorize the emotions. One theory, called discrete approach, seeks a set of basic emotions that can uniquely be distinguished (Plutchik and Kellerman, 1980). Examples include “interested”, “excited”, “upset”, “guilty”. The other approach groups emotions in major dimensions, which enable clear distinction among them (Russell, 1980). With this approach, emotions are mainly characterized by their valence, arousal, and recently, dominance. Valence (or pleasure) is the intrinsic attractiveness (or adverseness) of an event, object, or situation (Lewin, 1935). Arousal is the sensation of being mentally awake and reactive to stimuli, or a response to a difficult challenge where skills are acceptable. Dominance (or control, over-learning) is the sensation by which the individual skill level is higher than the challenge level for a task (Csikszentmihalyi, 1997a).

2.2.2 Affective States in Creativity, Problem Solving Skills, and Productivity

The literature reports that affective states have an impact on various cognitive activities of individuals (Khan et al., 2010). Creativity and cognitive processing are among them. According to Csikszentmihalyi (1997), creativity is something that distinguishes humans from animals. It is the process by which an individual uses symbols of a given domain and comes up with a new idea (or pattern)

that changes the domain itself. Creativity is a problem-solving action. Csikszentmihalyi distinguishes between convergent thinking and divergent thinking. Convergent thinking involves solving well-defined, rational problems that do not require significant creativity but an often unique, correct answer. Divergent thinking leads to no agreed upon solution and involves the ability to generate a great quantity of not necessarily correlated ideas (Csikszentmihalyi, 1997b).

Two recent meta analysis on creativity (divergent thinking) indicate a tendency to consider a positive mood as indicator of higher creativity. The studies in this field often classify moods as negative, neutral, and positive. In particular, positive mood enhances creativity with respect to a neutral mood, but there are not significant differences between the enhancements under positive mood and negative mood. Additionally, there are no significant differences between negative mood and neutral mood (Baas et al., 2008). Moreover, a positive mood enhances creative generation tasks while negative emotions do not offer enhancements (Davis, 2009). The work by Forgeard (2011) studies the relationship between mood, depression levels and creative generation and evaluation. Pre-existing, long-term mood moderates the effect of emotion inductions on creative thinking. More specifically, it raises the issue that inducing emotions to individuals has mild effect with respect their pre-existing mood. Additionally, participants low in depression have higher creativity under a negative mood induction (Forgeard, 2011). There are also findings that there are no differences between affective states and the quality of the creative work, but a positive mood increases the quantity of generated ideas (Sowden and Dawson, 2011).

Analytical problem-solving capacities - thus, related to convergent thinking - are believed to be higher when feeling negative emotions (Abele-Brehm, 1992, Melton, 1995). However, the work of Kaufmann and Vosburg (1997) shows that the participants with a positive mood performed non significantly better in analytic problem solving with respect to participants with negative affective states. On the other hand, the process of learning and cognitive processing are lowered by negative emotions (Brand et al., 2007).

Cognitive processing, especially analytical problem solving, and creativity are important for software development (Khan et al., 2010). It is natural to ask ourselves if long-term emotions are indicators of difference in creativity and problem-solving for software developers.

Affective states of the individuals have influence on work-related behaviors and capacities (Ilies and Judge, 2002). Additionally, the positive-Psychology branch defines the mental status of flow as completely focused motivation, energized focus, full involvement, and success in the process of the activity (Csikszentmihalyi, 1997a). The correlation with productivity looks straightforward. In fact, evidences have been found that happiest employees are more productive, regardless of the gender of the participants (Oswald et al., 2008, Zelenski et al., 2008). Moreover, the work by Fisher and Noble (2004) finds significant positive correlation between the productivity of different types of workers and positive emotions, and negative correlation between the productivity of workers and negative emotions (Fisher and Noble, 2004).

2.2.3 Measuring Affective States, Creativity, and Problem Solving Skills

The measurement of emotions is often achieved using surveys. The most famous survey for the discrete approach is the the Positive and Negative Affect Schedule (PANAS) (Watson et al., 1988). It is a 20-item survey that represents positive affects (PA) and negative affects (NA). The scale received many critiques. Among them, it does not always represent real feelings and tends to capture only high-arousal feelings in general (Diener et al., 2009). There are recent, modern proposals to further reduce the PANAS scale and improve it, bringing the discrete theory closer to the dimensional approach. The Scale of Positive and Negative Experience (SPANE) is a new proposal which has been validated to converge to the PANAS and other emotion measurements. It is composed by 12-items. Six items are related to negative experiences, while the other six are related to positive experiences. SPANE is based on the amount of time the feelings were experienced during the past 4 weeks and the final score, called affect balance score, represents an indication on the pleasant and unpleasant emotional feelings on the long term (Diener et al., 2009). One of the most used surveys for the dimensional approach is the Self-Assessment Manikin (SAM). SAM is a non-verbal assessment technique, based on pictures. SAM measures valence, arousal, and dominance associated with a person's affective reaction to a stimulus (Bradley, 1994).

To measure creativity is another open problem in Psychology. There are different tasks that aim to address this issue. The task that seems convenient to be reproduced by non-experts in the field is to generate captions for photographs. The participants receive random pictures of ambiguous meaning and write creative captions for them. A set of independent judges, experts in the fields of creativity, scores the captions (Forgeard, 2011). The scores for creativity are obtained from the number of generated captions (Sowden and Dawson, 2011), the average of the scores of all the captions, and the best caption written by each participant (Forgeard, 2011).

Measuring analytical problem solving skills is less tricky. There is only one solution to a given problem. Therefore, we can assign points to analytical tasks and determine the score for the tasks using the achievement made with respect to the task and the quantity of time spent on planning the solution. The Tower of London is a game aimed to determine impairments in planning and problem-solving capacities of individuals (Shallice, 1982).

2.2.4 Affective States and Software Development

The study by Khan et al. (2010) reports two empirical studies about affective states and software development. The studies are related to the developer's debugging performance with respect to the affective states. The results found that when the mood dimension of valence is kept high and the arousal dimension varies, there is a positive correlation with the debugging performance. This study requests for more research in the topic (Khan et al., 2010).

The study of Feldt et al. (2008) is a call for empirical software engineering studies using psychometrics. The authors argue that even though software is developed by humans, the research in software engineering misses to give importance to the humans themselves. The study uses personality tests on 47 participants to investigate a correlation with attitudes to software engineering processes and tools. The paper is about a work-in-progress research. They show that answers to questions are significantly different when the personality trait of “conscientiousness” changes. Conscientiousness is the tendency to think carefully before acting and to possess high needs for achievement. For example, higher levels of conscientiousness correlate with the preference to work alone, a low need to change working procedures, and to think that software engineering research is not important. In particular, the study calls for research about the role of the emotions in software engineering. The empirical software engineering branch that values software developers as individuals is tentatively entitled Individualized Software Engineering (Feldt et al., 2008).

Software Developers are employees and human beings. Their work mostly is cognitive. It is thus natural to ask ourselves if there is a correlation between the immediate productivity of software developers and their real-time emotions, where the stimulus is their immediate task.

3 Research Framework

Important links emerge from our review of the related work and the background research. In Section 1, we stated that developers are able to give general indications about their productivity. In sub-sections 2.1 and 2.2, we discovered that attempts to quantify the productivity of software developers are getting more and more complicated, but all fail when changing the context or miss empirical validations. The formulas incorporate several variables. Some of the variables include a quantification of the factors that influence the productivity. Still, something is missing. We believe that this “something” is related to the human aspect of software developers, more specifically, to their affective states.

3.1 Theoretical Model

The body of knowledge from Psychology and Cognitive Science reports several connections between affective states, important skills required for software development, and productivity of the individuals.

Based on the literature review, the lack in considering the human aspect of software developers, and the call to the usage of psychometrics in individualized software engineering, we build our research framework that supports research in empirical software engineering through psychometrics. There are at least two different approaches when developing theoretical models: variance (or factor) theory and process theory. Experimental methods usually adopt a variance approach to causal inference. Researchers deduct the causes of events from the theory and research designs in which the presence of the cause is demonstrated by a systematic relationship between inputs and outputs (Morris, 2005). Process theories provide explanations by studying sequences of events that lead to outcomes, where the sequences of events are not likely to be linear (Langley, 1999).

Even if we are aware that process theories for theoretical models are gaining more and more attention in research fields, we believe that they are more appropriate for data collected in real organizational context (e.g. case studies). Therefore, we stick with the classic variance theory without demanding to demonstrate causalities.

The affective states related to software development skills are long-term feelings. The productivity related to a developer’s task is real-time and the same holds for the mood of the task stimulus.

For these reasons, we created two separate theoretical models that we implement in two empirical studies. The first study is a research about long-run correlation of affective states on skills for software development. The second study is about real-time correlation of the mood on immediate productivity of the development task. We report the theoretical models in Figure 1 and in Figure 2.

From the related work, it emerges that there is a correlation between affective states, creativity, and analytical problem-solving. However, there are many contradictory studies to claim that this correlation is either positive or negative.

For this reason, Model 1 contains unsigned relationships. The empirical evidence provides insights on the sign of those relationships.

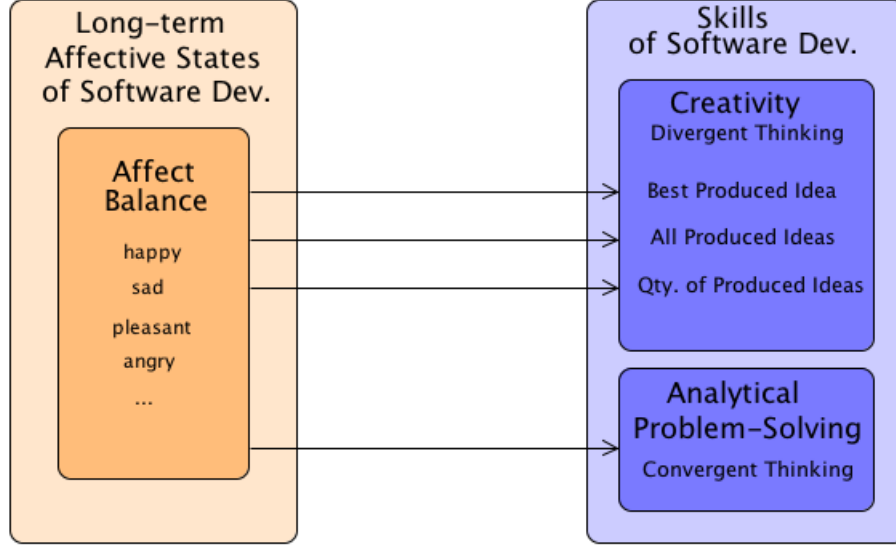


Figure 1: Model for Study 1

Regarding Model 1 in Figure 1, we draw the independent variables on the left side (orange boxes) and the dependent variables on the right side (blue boxes). The independent variables are related to long-term affective states and are represented by the Affect Balance variable when using the SPANE measurement instrument. The dependent variables are related to skills of software developers when solving problems. For the creativity (divergent thinking), we consider as variables the value of the best produced idea, the average value of all the produced ideas, and the quantity of the produced ideas to solve a problem. For the analytical problem-solving (convergent thinking), we will define a suitable metric in the task to be solved (section 5.1.4.6).

The body of knowledge reports that affective states are positively correlated to the productivity of individuals. Therefore, our research hypotheses in Model 2 seek for positive correlations between real-time affective states and immediate productivity of software developers.

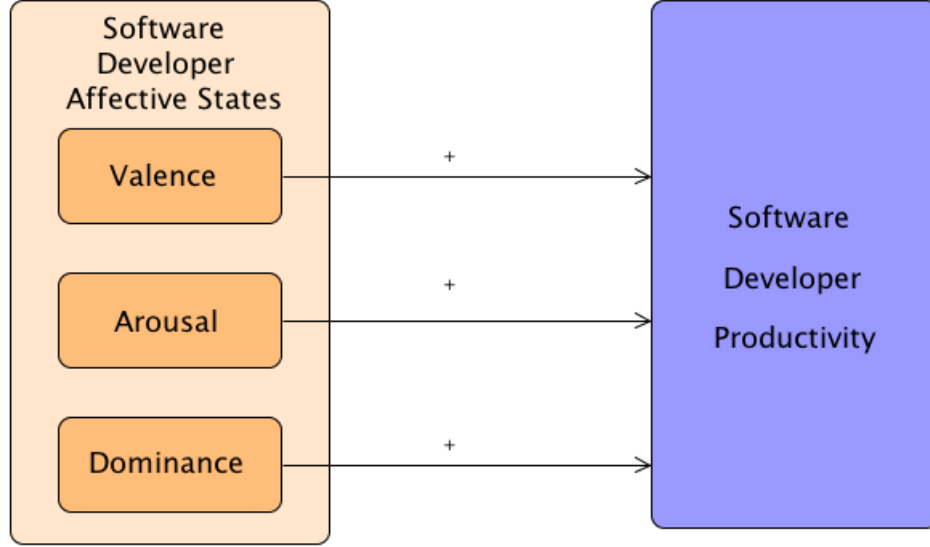


Figure 2: Model for Study 2

In Model 2 (Figure 2), we draw the independent variables on the left side (orange boxes) and the dependent variables on the right side (blue boxes). The independent variables are immediate affective states related to the software development task and are the dimensions of valence, arousal, and dominance. The dependent variable is productivity and will better be defined in our empirical study, section 5.2.3.4.

3.2 Research Hypotheses

Our theoretical models lead to the creation of 7 Research Hypothesis (RH). The following are the RH of this thesis, formulated in their null hypothesis form.

- H1₀: There is no difference in the best creativity between software developers feeling positive affective states and software developers feeling non-positive affective states.
- H2₀: There is no difference in the average creativity between software developers feeling positive affective states and software developers feeling non-positive affective states.
- H3₀: There is no difference in the number of generated creative solutions between software developers feeling positive affective states and software developers feeling non-positive affective states.

- H4₀: There is no difference in analytical problem solving skills between software developers feeling positive affective states and software developers feeling non-positive affective states
- H5₀: There is no correlation with the valence mood dimension and the productivity of software developers
- H6₀: There is no correlation with the arousal mood dimension and the productivity of software developers
- H7₀: There is no correlation with the dominance mood dimension and the productivity of software developers

Even if we search for a difference in H1, H2, H3, and H4, we look for a positive difference between participants feeling positive affective states and the skills of divergent thinking and convergent thinking. Nevertheless, we use two-tailed hypotheses, because the literature review does not always provide clear indications. These four hypotheses are addressed in Study 1, section 5.1.

Hypotheses H5, H6, and H7 are related to real-time performance of software developers and their mood dimensions. Based on the review, we look for positive correlations. These three hypotheses are addressed in Study 2, section 5.2.

4 Empirical Research Design

According to Wohlin et al. (2000), empirical research is mainly carried out through three types of investigations, namely surveys, case studies, and experiments.

Surveys are often performed in retrospect to events or the usage of certain techniques and tools. Surveys include questionnaires and interviews, in which the first is mainly used to gather and analyze quantitative data.

Case studies are observational experiences for monitoring projects, activities or assignments. Data is collected by the researcher without interfering with the activities carried on by the participants.

Experiments provide a high level of control and are usually carried on in a controlled environment, like a laboratory. Often, participants are randomly assigned to different treatments, to represent the different conditions (e.g. tools, techniques) to be tested. With experiments, participants under different treatments are assigned to different groups. The inference analysis of the data is to compare the different treatments (i.e., groups) in order to find significant evidences of which treatment is the best one (Wohlin et al., 2000).

Therefore, both our studies are controlled experiments.

The first one is to be performed in a controlled environment - a computer laboratory. The participants individually perform the tasks and their results are evaluated in order to test the hypotheses.

The second study is unusual for the field of software engineering. In SE we can find analysis of repeated measures designs, in which all the participants perform the same task and are measured either after a determined interval of time or after certain events happen. Our study differs from this approach, because all the participants perform a different task and are individually studied for a determined period. Their affective states and their productivity are measured at equal intervals of time, but the same interval numbers (e.g., interval 1 for participant P1 and interval 1 for participant P2) have no correlation among them. This approach is an hybrid between repeated measures design and the Experience Sampling Method (ESM) (Larson and Csikszentmihalyi, 1983). In ESM designs, the participants stop at certain times and make notes of their experience in real time. They often record temporal things like feelings and thoughts in that precise moment.

The mostly adopted instrument to model and analyze repeated measures with particular dependencies among data and time, or ESM designs, is a linear mixed effects model (Laird and Ware, 1982). It is a linear model that contains both fixed effects and random effects. We provide a general definition of a linear mixed effects model in section 5.2.3.5.

As a side note, we do not induce emotions on the participants of our studies because pre-existing mood mitigates the effect of mood induction (Forgeard, 2011). Additionally, we would like to analyze pure affective states that individuals feel, without interfering with them.

5 Empirical Results

This section of the thesis provides our two studies to test the research hypotheses. They are reported as almost self-contained research papers. We wrote the papers following the guidelines of Wohlin et al. (2000) and Jedlitschka et al. (2008). We analyze the effects of pre-existing, long term affective states and software development skills in Section 5.1, Study 1. We analyze the correlation between real-time mood dimensions and the productivity of software developers in Section 5.2, Study 2.

5.1 Study 1: Affective States and Software Development Skills

Abstract

Background: software engineering lacks studies about the role of affective states for important software development skills, such as analytical problem solving and creativity.

Objective: Provide a quantitative evaluation of the benefits of pre-existing affective states on the creativity and problem-solving capacities of Software Developers.

Method: Controlled, randomized experiment with N=42 students of Computer Science (B.Sc & M.Sc). Two tasks, one related to creativity and the other one related to problem-solving capacity.

Results: Evidences that positive pre-existing affective states are indicators of higher problem solving capacities ($p=0.0079$). No evidences that positive pre-existing affective states are indicators of higher creativity. However, the data tendencies suggest a correlation to be addressed in future studies.

Limitations: Specific population analyzed (Computer Science students) and limited in the age. Too few participants under extremely negative affective states.

Conclusions: The results provided evidences that software developers feeling more positive affective states possess higher problem solving capacities. More studies are needed for the meaningful results for the creativity. More studies are needed to analyze extremely negative affective states.

5.1.1 Introduction

In section 1 and section 2, we pointed out the need to study the role of software developer’s affective states. In particular, two important skills required for the whole area of software development are analytical problem-solving and creativity.

In this study we report a controlled experiment to research the effects of pre-existing affective states on the creativity and the problem-solving skills of software developers. We created two tasks, supported by related work from Psychology and Cognitive Science, in which we assess these two factors while controlling the affective states of the participants.

We performed the experiment in a controlled environment with 42 students of Computer Science, diverse in country, age, and level of study.

Our results indicate that for the problem-solving skill there are significant differences between participants facing the most positive emotions and those facing less positive emotions. We do not find significant evidences for the creativity, but the data trends may indicate a correlation, to be discovered in future studies. Additionally, we ask to future studies to address the problem of finding more participants feeling negative emotions.

The impact of our results may lead to research in enhancements of the emotions felt by students of Computer Science, and software developers, in order to boost their capacities.

5.1.1.1 Motivation The purpose of our study is to provide a quantitative evaluation of the benefits of pre-existing affective states on the creativity and problem solving capacities of software developers.

Even if a body of knowledge exists, it is still difficult to define programming tasks to assess such skills, while controlling individual related variables (e.g., experience with the programming language) and gather a significant number of participants.

In order to provide meaningful results, we abstract the process by adopting Psychology-related tools and tasks. To this end, we define two tasks to be sequentially performed in a controlled environment (i.e., a computer laboratory). The first task is related to the creativity of the participants, while the second task is related to their problem-solving capacities. For each of the two tasks, the experiment compares the performance of two groups of participants: those feeling positive affective states vs. the remaining participants (neutral and negative affective states), according to a median split.

In section 1.3, we stated the main goal of our thesis. This work provides the first answers to the research sub-questions 1 and 2:

1. How do pre-existing affective states influence the creativity of software developers?
2. How do pre-existing affective states influence the analytical problem-solving skill of software developers?

We transform these two research questions in research objectives and hypothesis in the Goal Definition section.

5.1.2 Related Work

In the previous section we stated the motivations of our research and introduced our research questions. While in software engineering there are no studies providing insights to our research questions, there is a rich body of knowledge in Psychology about affective states and creativity. We reported the related work about affective states and creativity in sections 2.2.1 and 2.2.2. In this section we write about the related studies that provide empirical evidences. We will compare our results with these results in section 5.1.6 (Discussion).

The work by Forgeard (2001) adopts mood induction, measures depression levels and assesses creative evaluation and generation. We implemented the same creative generation task and the same assessment technique of the captions. Among other findings, this study concludes that participants low in depression perform better in the creative generation task with an induced negative mood. Additionally, mood induction techniques have tiny effects with respect to pre-existing affective states (Forgeard, 2011).

On the other hand, the study by Sowden and Dawson (2001) finds that there are no differences between affective states and the quality of the creative work. Additionally, positive mood increase the quantity of generated ideas (Sowden and Dawson, 2011).

Regarding problem-solving skills, the majority of the studies agree that individuals feeling negative emotions perform better in analytical tasks (Abele-Brehm, 1992, Melton, 1995). Nevertheless, the work of (Kaufmann and Vosburg, 1997) shows that the participants with a positive mood perform better in analytic problem solving with respect to participants with negative affective states, although non significantly.

5.1.3 Description of the Experiment

In the previous section we wrote about the work related to our research questions. In this section we describe our controlled experiment in order to provide the necessary information for replication.

We define the goal of our study following the guidelines of Wohlin (2000). From the research goal, we derive the research objectives that lead to the formulation of the research hypothesis in the Experimental Design subsection. After the formulation of the hypothesis, we describe all the variables involved in the analysis of the data, the participants and the instrumentation. We then provide detailed information on the two tasks that the participants perform in the Execution sub-section.

5.1.3.1 Goal Definition The **object of study** is Computer Science students and their ability in terms of creativity and problem solving skills.

The **purpose** of the experiment is to evaluate the individual skills based on pre-existing positive or non-positive affective states. The experiment provides insights in what we can expect in terms of individual skills when controlling the affective states.

The **perspective** is from the point of view of the researchers. The researchers want to know if there is any systematic difference in the individual skills based on pre-existing affective states. The main effect studied in the experiment is the individual skills in two tasks related to problem solving and creativity.

Two specific aspects are emphasized for the **quality focus**. We focus on creativity and problem solving scores provided by two specific tasks. The creativity score is a mark given by a set of independent judges. The problem-solving score is a ratio between the Tower of London game score and the sum of planning strategy time. We provide more details about the variables in the next section.

The **context** of the controlled experiment is a laboratory environment (computer laboratory). M.Sc. and B.Sc. students in Computer Science perform the tasks. The ability to generalize from this specific context is further elaborated in the threats to validity section.

The experiment addresses a real problem, i.e. the difference in individual performance while controlling the affective states. The use of a computer laboratory as an experimental context provides excellent opportunities to replicate the experiment. The required material per participant is one personal computer, a pen and printed handouts. We further discuss the materials needed in the Experimental Design section.

We use the goal-question-metric model (Basili and Weiss, 1984, Jedlitschka and Ciolkowski, 2008) to define the goals, here summarized.

O₁ ANALYZE THE CREATIVITY (DIVERGENT THINKING)
 FOR THE PURPOSE OF EVALUATION
 WITH RESPECT TO PRE-EXISTING AFFECTIVE STATES OF THE INDIVIDUALS
 FROM THE POINT OF VIEW OF THE RESEARCHER
 IN THE CONTEXT OF M.SC. AND B.SC. STUDENTS WRITING CAPTIONS
 FOR PHOTOGRAPHS.

O₂ ANALYZE THE PROBLEM SOLVING SKILLS (CONVERGENT THINKING)
 FOR THE PURPOSE OF EVALUATION
 WITH RESPECT TO PRE-EXISTING AFFECTIVE STATES OF THE INDIVIDUALS
 FROM THE POINT OF VIEW OF THE RESEARCHER
 IN THE CONTEXT OF M.SC. AND B.SC. STUDENTS PLAYING THE TOWER
 OF LONDON GAME. (Shallice, 1982)

5.1.4 Experimental Design

This study concerns a controlled experiment to analyze skills and affective states of software developers. The experiment does not aim to simulate a working

environment. Computer Science students are randomly chosen to perform the tasks in a computer laboratory. In this sub-section we provide information in order to replicate the experiment and to evaluate the internal validity of the experiment.

5.1.4.1 Hypothesis Formulation Software developers are human beings who feel affective states. As human beings, their affective states influence their performances. We claim that there are differences in the creativity and the problem-solving skills with respect of the affective states.

While affective states can also be generalized in three groups (negative, neutral, and positive), we group them in two categories, namely POS and N-POS, using a median split.

We derive the hypothesis from the Research Objectives defined in the previous section and the Theoretical Model 1 (Figure 1) of section 3.1.

1. Null hypothesis, H1₀: There is no difference in creativity between software developers feeling positive affective states and software developers feeling non-positive affective states.

$$H1_0: \mu_{BCreat_{N-POS}} = \mu_{BCreat_{POS}}$$

$$\text{Alternative Hypothesis, } H1_1: \mu_{BCreat_{N-POS}} \neq \mu_{BCreat_{POS}}$$

Measures needed: Software developers affect balance score and best creativity score

2. Null hypothesis, H2₀: There is no difference in the average creativity between software developers feeling positive affective states and software developers feeling non-positive affective states.

$$H2_0: \mu_{Creat_{N-POS}} = \mu_{Creat_{POS}}$$

$$\text{Alternative Hypothesis, } H2_1: \mu_{Creat_{N-POS}} \neq \mu_{Creat_{POS}}$$

Measures needed: Software developers affect balance score and average of the creativity scores

3. Null hypothesis, H3₀: There is no difference in the number of generated creative solutions between software developers feeling positive affective states and software developers feeling non-positive affective states.

$$H3_0: \mu_{num.captions_{N-POS}} = \mu_{num.captions_{POS}}$$

$$\text{Alternative Hypothesis, } H3_1: \mu_{num.captions_{N-POS}} \neq \mu_{num.captions_{POS}}$$

Measures needed: Software developers affect balance score and the number of creativity solutions.

4. Null hypothesis, H4₀: There is no difference in problem solving (0.00 to 1.00 score) between software developers feeling positive affective states and software developers feeling non-positive affective states

$$H4_0: \mu_{ProbSolv_{N-POS}} = \mu_{ProbSolv_{POS}}$$

$$\text{Alternative Hypothesis, } H4_1: \mu_{ProbSolv_{N-POS}} \neq \mu_{ProbSolv_{POS}}$$

Measures needed: Software developers affect balance score and problem

solving score

5.1.4.2 Variables We now describe the variables of the experiment.

The Affect Balance Score is obtained from the Scale of Positive and Negative Emotions (SPANE) (Diener et al., 2009). It is a 12-item scale, which assesses the frequency of 6 positive and 6 negative affective states during the last 4 weeks. By subtracting the sum of the negative items frequencies from the sum of the positive items frequencies, we obtain the affect balance score (SPANE-B). SPANE-B ranges from -24 (completely negative) to +24 (completely positive), where 0 is a completely neutral score. SPANE-B is a fuzzy indication of the mood of individuals. Therefore, we split the participants using the median of the SPANE-B score.

The creativity score (Creat) is obtained from the instructions provided by Forgeard (2011). Participants write as many captions as they want for the photographs. A committee of independent judges - experts in the creativity field - ranks the captions. The participation of the judges is volunteer. Each judge assesses all captions using a mark that ranges from 1 to 7, with 1 representing no creativity at all. We assign to each caption the average mark of the judges. Therefore, the Creat score is the average mark of a caption. For each participant, we consider the average Creat score of their captions, and the number of generated captions. Additionally, we retain the best Creat score (BCreat), because creators are often judged by the quality of their best works rather than by the quality of all works averaged (Simonton, 1997). The instructions that we provide to the judges can be found in Appendix B.

The problem solving score (ProbSolv) is a metric that we derive from two different measures. The first measure is the Tower of London Score Sum (TOLSS). It describes the ability to solve 12 different problems with maximum three trials to solve each one. If a problem is solved at the first trial, 3 points are added to TOLSS. If a problem is solved at the second trial, 2 points are added to TOLSS. Consequently, 1 point is added to TOLSS if the problem is solved at the last possible trial and 0 otherwise. With 12 problems to be solved, TOLSS ranges between 0 and 36. The second measure is the Planning Time Sum (PTS). It describes the sum of the seconds spent to plan the solutions of all the problems. For each trial, we measure the seconds passed from the presentation of the problem to the beginning of the implementation of the solution. This translates to the seconds passed from the presentation of the problem to the first mouse click in the program. Finally, we define ProbSolv as $\frac{TOLSS}{PTS}$. That is, we use PTS as either a bonus or a penalty for TOLSS. A participant who solves all the problems at the first trial and thinks less time is considered better than a participant who solves all the problems and thinks more time. The ProbSolv ratio has a range that varies depending on the participants. In order to have results that are easier to be understood and compared, we use a function to map ProbSolv to a range from 0.00 to 1.00. The mapping formula is $f(x) = \frac{x - \min(ProbSolv)}{\max(ProbSolv) - \min(ProbSolv)}$.

We provide a better understanding of the scores and the survey in the *Objects*

and *Instrumentation* paragraph, and in the *Execution*, and *Analysis* sections.

5.1.4.3 Participants Units A total of 42 participants was sampled from the Faculty of Computer Science students of the Free University of Bozen-Bolzano. The participation was voluntary and in exchange of credit-points. Of the 42 participants, 33 were male and 9 female. The sample had a mean age of 21.50 years old (SD=3.01 years) and was diverse in provenience Country: Italy 74%, Lithuania 10%, Germany 5%, and Ghana, Nigeria, Moldavia, Peru, U.S.A. all with a 2.2% frequency. Additionally, we recorded the number of years of study, with a mean of 2.26 years (SD=1.38 years).

5.1.4.4 Objects and Instrumentation The objects required for the creativity task consists of 6 color photographs with somewhat ambiguous meanings³. Three of the photographs of our experiment were provided by Forgeard (2011). The remaining three were provided by a Design and Arts Professor of the Free University of Bozen-Bolzano. Each participant receives two printed pictures.

The material required for the problem-solving task is provided by the open-source PEBL software (Piper et al., 2011). One computer per participant is required.

For the affective state measurement we adopted a web form that implements the Scale of Positive and Negative Emotions survey (SPANE) (Diener et al., 2009). Our implementation of the SPANE survey can be found in Appendix D.

We now describe the instrumentation that we used to measure the variables.

For the creativity task, we created a spreadsheet with the generated captions to be distributed to a committee of expert judges (Forgeard, 2011, Kaufman et al., 2007).

For the problem solving task, the PEBL software automatically collects all the required data.

5.1.4.5 Validity Evaluation In order to minimize the bias related to the knowledge of the experiment procedure, we only asked the students to participate to a study for a duration of maximum two hours. To further assure anonymity, we assigned them a random participant code, to be used as a reference through the stages of the experiment. The first task can induce effects to the mood of the participants. Therefore, we measured the affective states before each task. The Cronbach’s Alpha (Cronbach, 1951) reliability score of the two SPANE-B scores of the participants is $\alpha = 0.971$, indicating a strong reliability of the data. We did not induce emotions on participants. Therefore, they happened to have the pre-existing affective states that they felt. In this sense, true randomization happened. The two photographs were randomly assigned, one at a time. We only asked the judges to participate to an experiment about

³For copyright reasons, the six photographs are not included here, but are available from the author upon request.

creativity. Additionally, the judges received the captions grouped by picture, without any indication about the authors of the captions.

The problem solving task was the same for all the participants. We assigned the participants to the two groups using a median split. The control of the affect balance score was determined by the results of the SPANE survey. Therefore, we assigned the participants to the POS and N-POS groups after the execution of the experiment. For this reason, the participants did not know about the existence of the groups.

Besides the collection of the written captions, all the measurements were automatized.

5.1.4.6 Execution In this section we describe the two tasks performed by the participants.

Creative Generation Task The participants received two random photos from the set of the 6 available one, one at a time. They were provided with the instructions in Appendix A. Please note that the instructions given to participants are the same of the task executed by Forgeard (2011). They were given no time restrictions to generate creative captions for the photo received. The captions were all written in English. The participants were non-native English speakers. Nevertheless, the Free University of Bozen-Bolzano is a trilingual school. The Faculty of Computer Science holds the lectures in English and a requirement in order to enroll for the University is the acquisition of an English certificate. Nevertheless, they were allowed to use on-line translators and dictionaries. The task was completed in thirty minutes.

Problem-solving Task The participants opened the PEBL software. The test battery was already setup to open the appropriate test, namely *Shallice test* (*[1,2,3] pile heights, 3 disks, Shallice's 12 problems*). Figure 3 is a screen-shot of the first round of the test. We report the instructions of the game in Appendix C. Please note that the same instructions are included in the PEBL software. The software displays the instructions before the start of the task. The task was completed in thirty minutes.

5.1.5 Analysis

In the previous section we provided detailed information for the reader in order to understand and eventually replicate our experiment.

In this section we describe the performed analysis, in order to obtain meaningful results for our Research Goals and the Research Hypothesis. For both the two tasks, we provide descriptive statistics about the distribution of the SPANE-B affect balance score. Then, we provide and discuss the box-plots of the Creat and the ProbSolv scores for the N-POS and POS groups. Then, we discuss the reductions to the datasets and the reasons behind them. Lastly, we

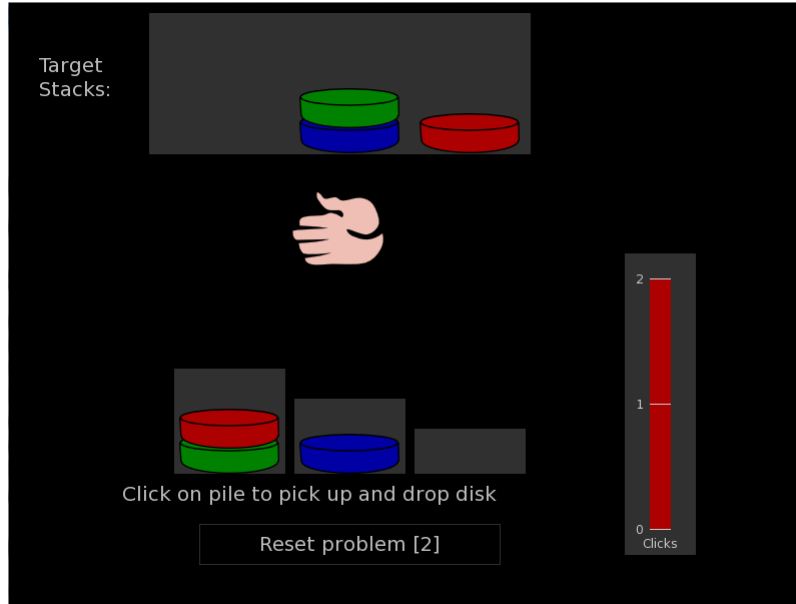


Figure 3: Tower of London Game

provide our test for the research hypothesis and the results of the statistical tests to achieve them.

5.1.5.1 Descriptive Statistics As a first step to analyze the data, we provide descriptive statistics.

Creativity Task

All the 42 participants completed the task and provided valid data for the SPANE survey. One participant was excluded from the data-set, see the *Data Reduction* sub-section. We report the classification of the mood in Figure 4.

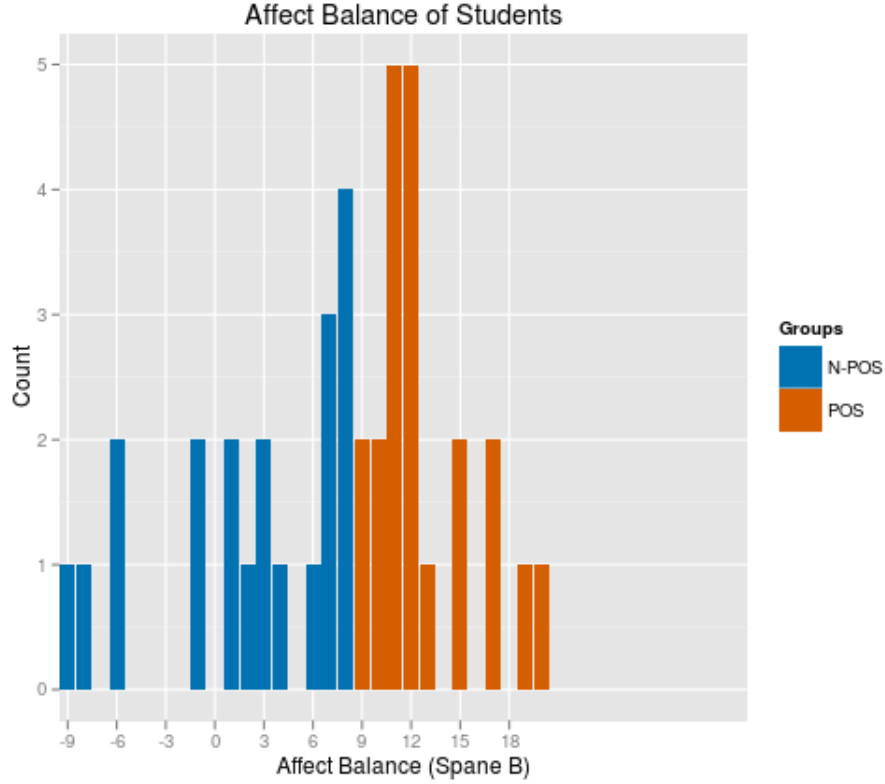


Figure 4: Affect Balance Score for the Creativity Task

The average for the SPANE-B was 7.58 (Standard Deviation=7.04) and the median was 9. 20 participants were classified as N-POS and 21 of them were classified as POS. We note that the N-POS group only contains one person with a SPANE-B less than -8. That is, the participants in N-POS are all between neutral and slightly negative in their pre-existing affective states.

The participants produced a total of 220 captions. With an average of 5.24 captions per participant, we agree that they enjoyed the task. It is important to observe the behavior of the participants as additional validation of the SPANE-B results. In Figure 5 we provide the box-plots for the number of generated captions of the two groups. The N-POS group wrote in average 4.70 captions (SD=2.34), while the POS group wrote an average of 5.90 captions (SD=3.47). We note in Figure 5 that the POS group has a higher maximum number of generated captions (13), as well is higher the median of the written captions.

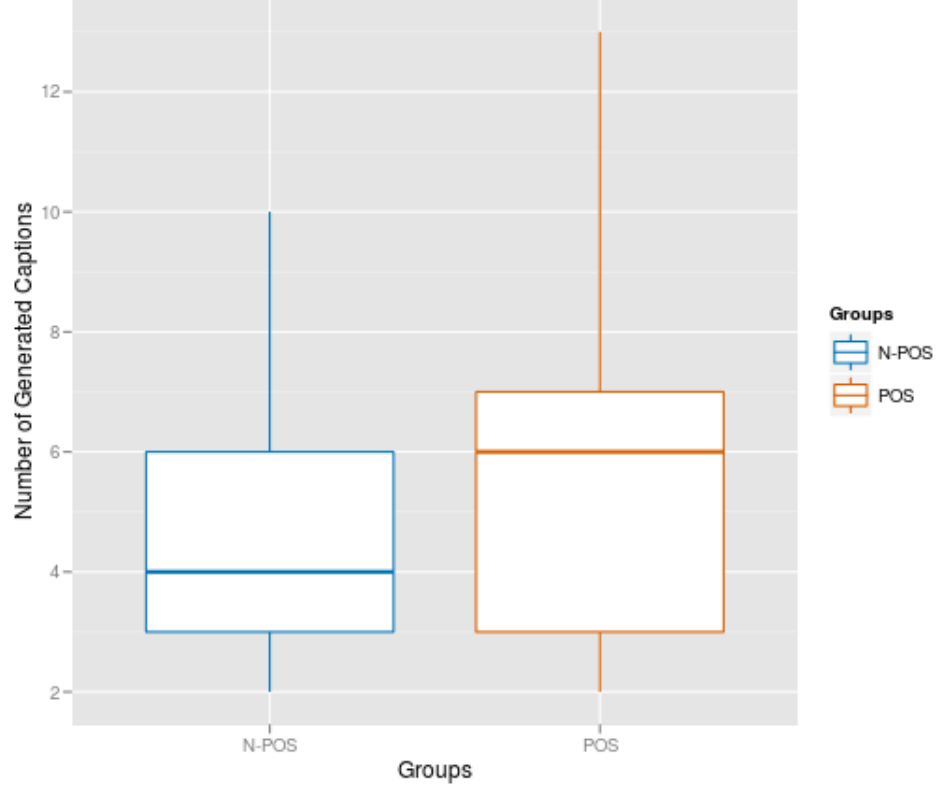


Figure 5: Box-plots for the number of generated ideas of N-POS and POS groups

The average of the BCreat score was 4.00 (SD=0.75). In Figure 6 we provide the box-plot of the BCreat score for the two groups. The average BCreat score of the N-POS group was 4.02 (SD=0.76). The average BCreat score for the POS group was 3.98 (SD=0.76). As we can see in Figure 6, there are no noticeable differences among the two groups. The scatter-plot in Figure 7 does not provide evident trends in the data, but the graphical best fitting lines may indicate interesting results that will be discussed in the next sections.

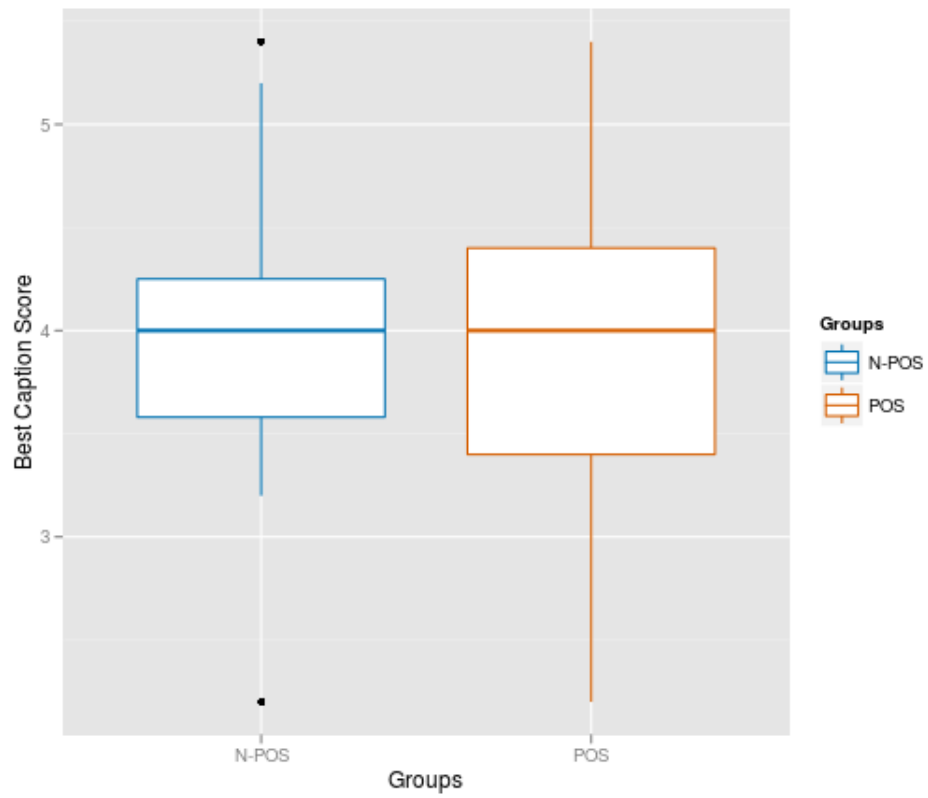


Figure 6: Box-plots of the BCreat score for the N-POS and POS groups

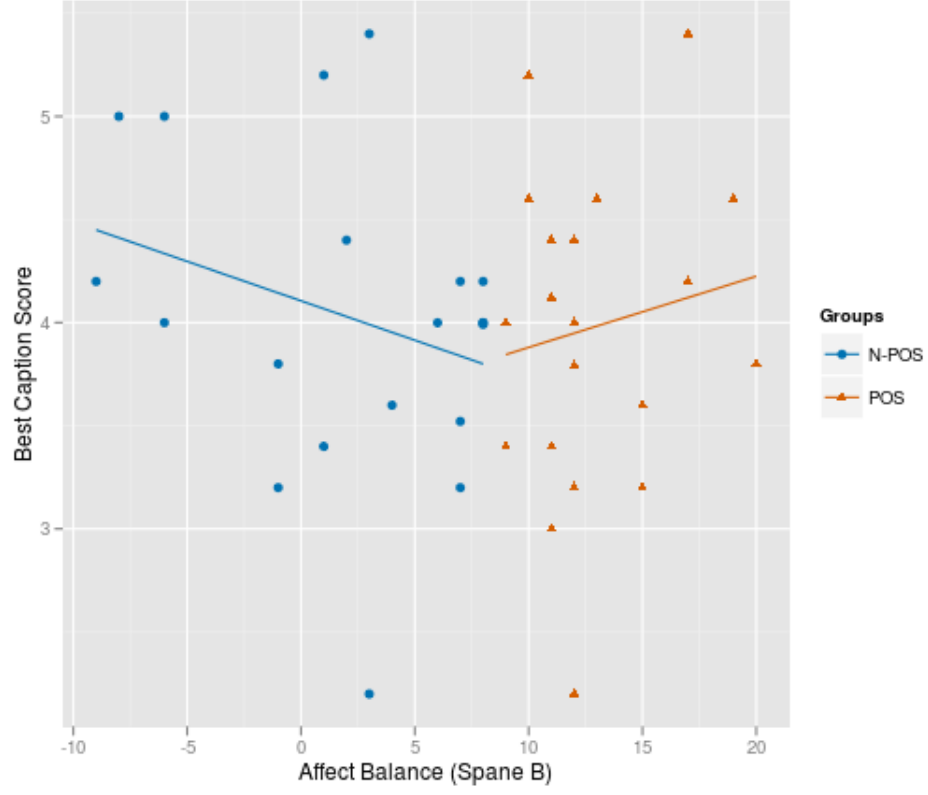


Figure 7: Scatter-plot BCreat score vs. SPANE-B

The average Creat score was 3.11 (SD=0.59). The N-POS group averaged a score of 3.13 (SD=0.46), while the POS group averaged a score of 3.08 (SD=0.59). In Figure 8 and Figure 9 respectively, we provide the box-plots and the scatter-plot of the groups. Again, the graphical trends of the data will be discussed next.

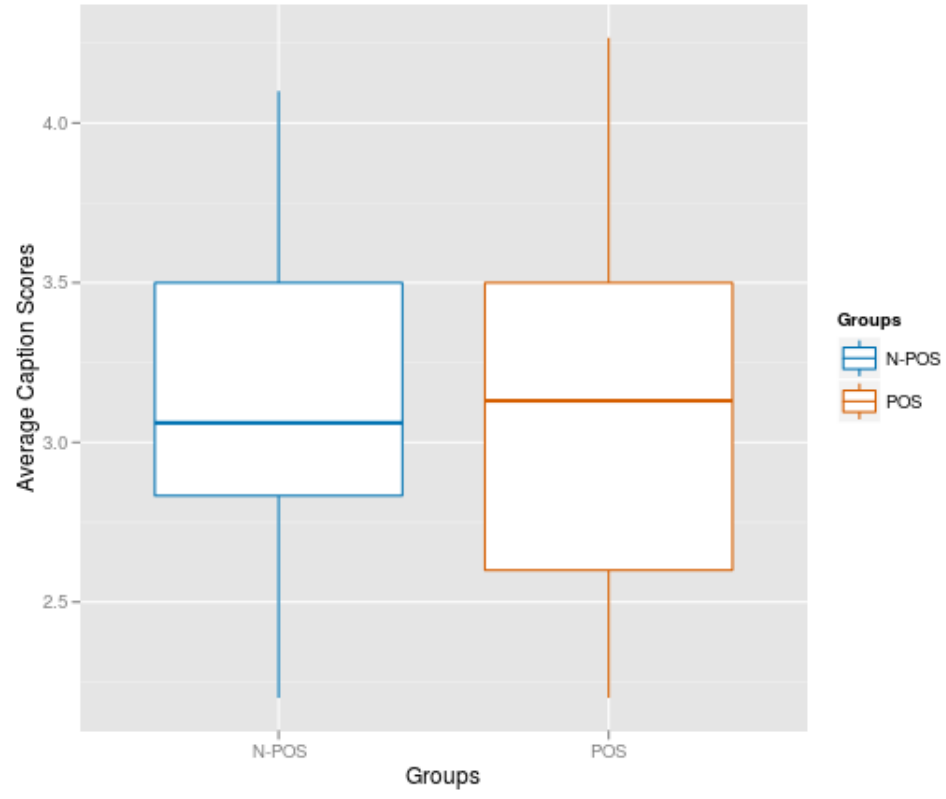


Figure 8: Scatter-plot μ_{Creat} for the N-POS and POS groups

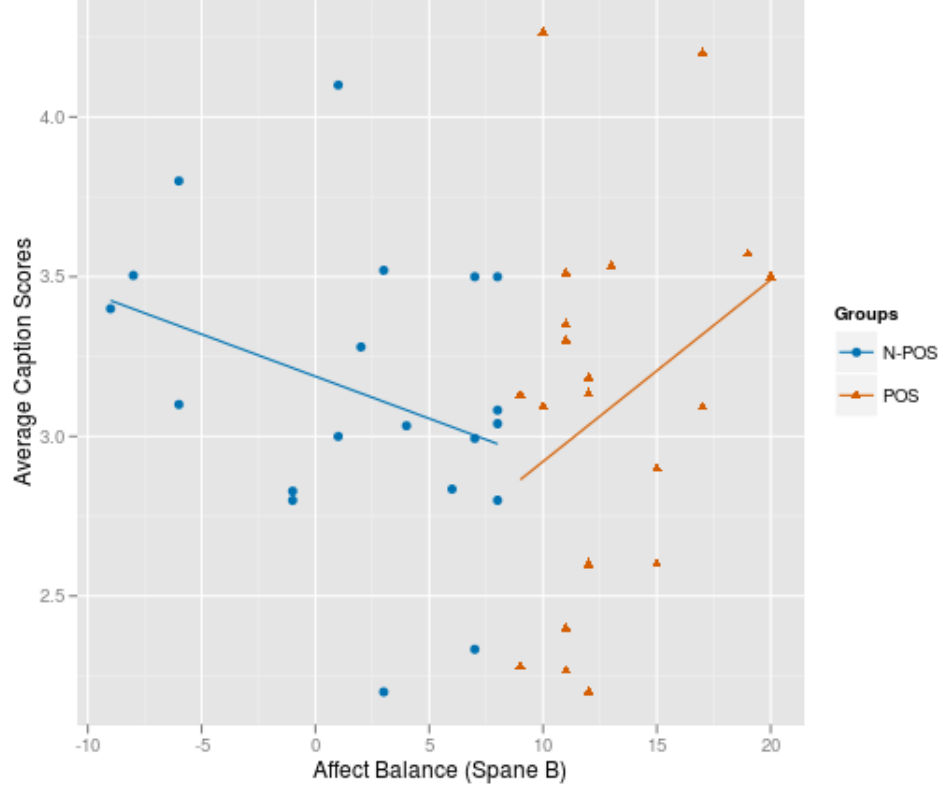


Figure 9: Scatter-plot μ_{Creat} vs. SPANE-B for the N-POS and POS groups

Problem-Solving Task

41 participants provided complete data for the SPANE survey. One participant was excluded from the data-set because of missing data.

The SPANE-B average score was 7.58 (SD=6.69), while the median was 10. We report in Figure 10 the distribution of the mood after the Creativity Task. With 19 participants classified as N-POS and 22 participants classified as POS, we note that the N-POS group does not contain participants with a SPANE-B less than -8. That is, the participants in N-POS are all between neutral and slightly negative in their pre-existing affective states. Additionally, we report a weak mood-induction effect caused from the first task. Nevertheless, the Cronbach's Alpha (Cronbach, 1951) reliability score of the two SPANE-B scores of the participants is $\alpha = 0.971$, indicating a strong reliability of the data.

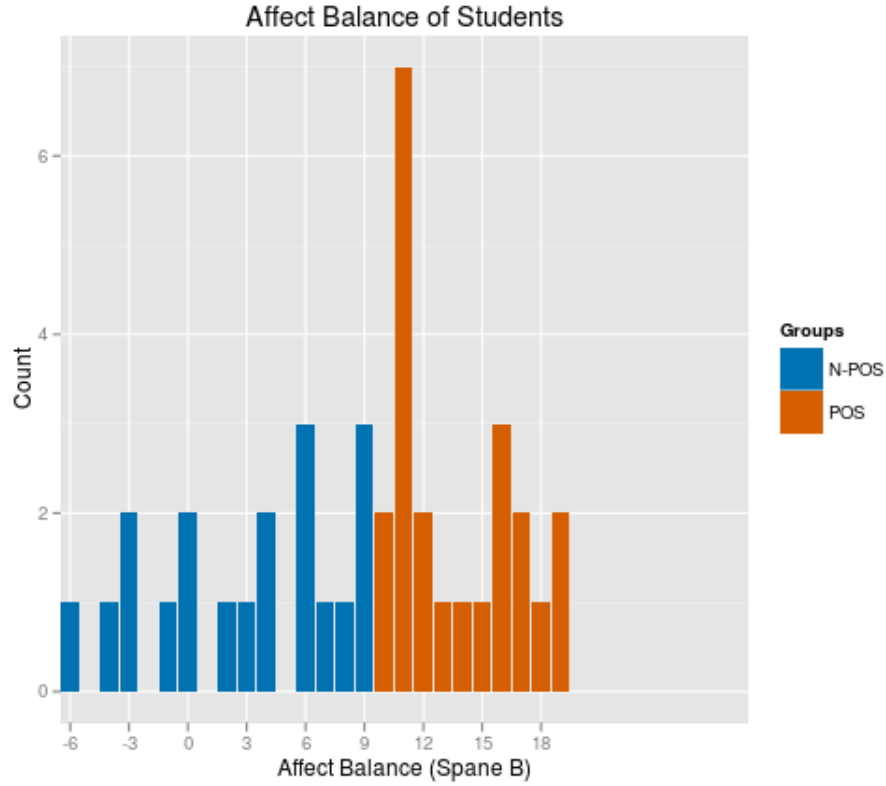


Figure 10: Affect Balance Score for the Problem-Solving Task

The ProbSolv average score was 0.18 (SD=0.08). The average ProbSolv score for the N-POS group was 0.14 (SD=0.05). The average ProbSolv score for the POS group was 0.20 (SD=0.09). We see in the box-plots in Figure 11 that there appears to be a difference between the two groups. The POS group has a higher average score and the distribution around the average score is right-skewed, suggesting excellency scores. The scatter-plot in Figure 12 further confirms this supposition. It appears that the ProbSolv score decreases as the SPANE-B increases for the N-POS group. The tendency seems reversed for the POS group: the ProbSolv increases non-linearly as the SPANE-B increases. Only 3 participants for the N-POS group scored above 0.2 and 0 of them scored above 0.3. 10 participants for the POS group scored above 0.2 and 4 of them scored above 0.3.

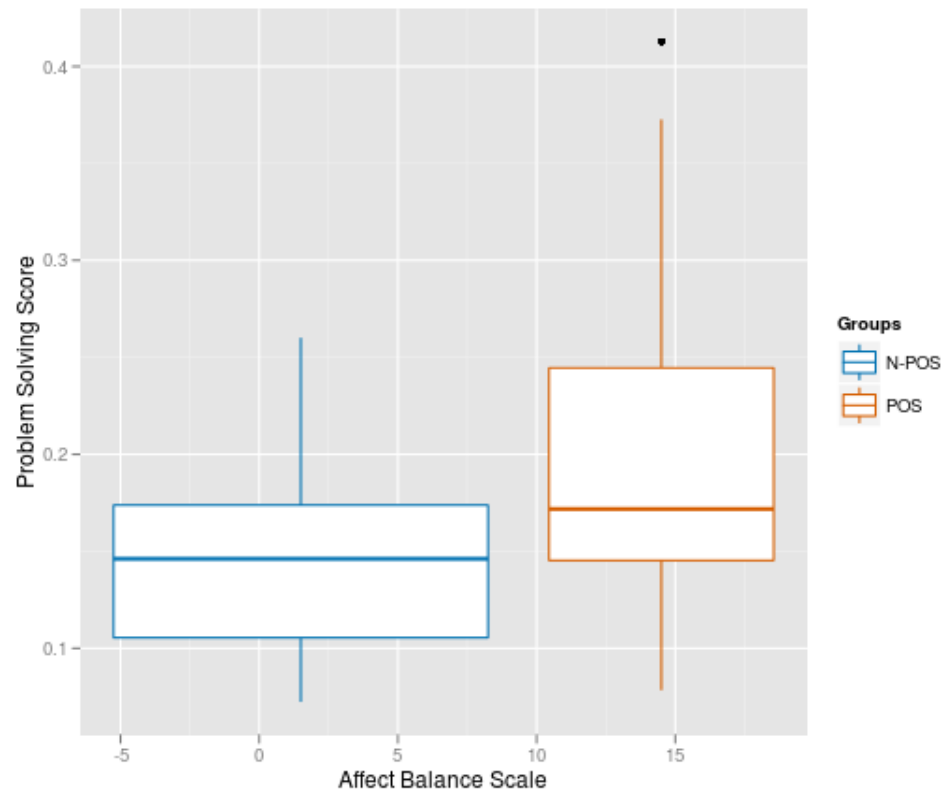


Figure 11: Box-plots of the ProbSolv score

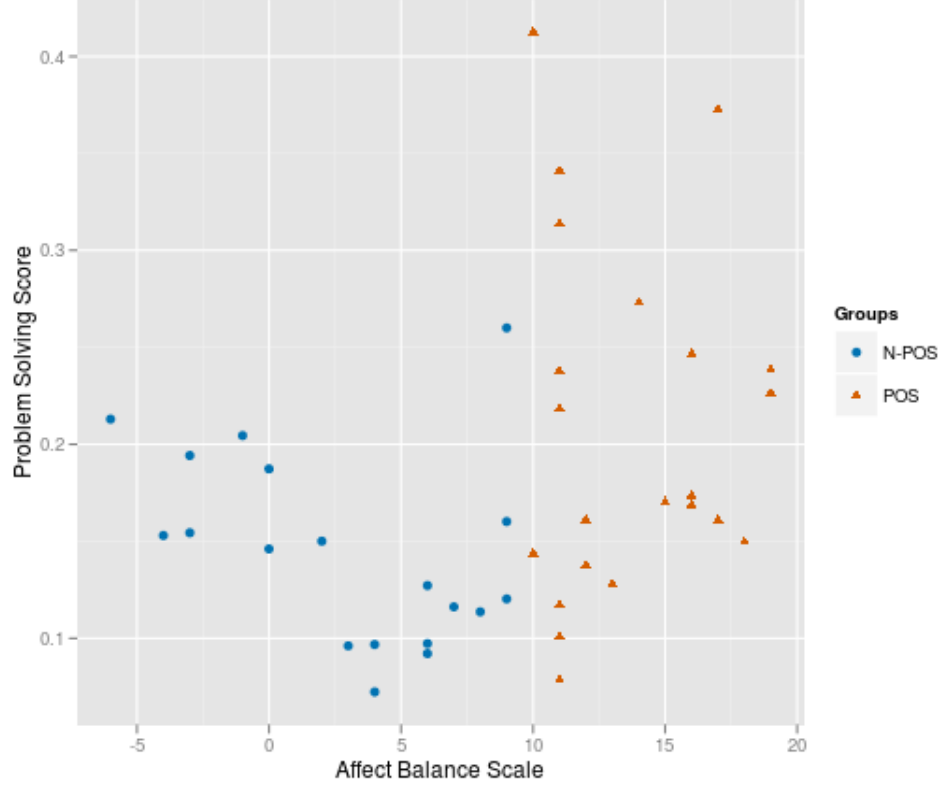


Figure 12: Scatter-plot Creat vs. SPANE-B

5.1.5.2 Data Reduction For the Creativity task, one participant obtained a maximum score of 1.40. With such a low score, we investigated the provided captions. The participant wrote one caption per photograph. Each caption literally described the content of the photograph. Therefore, we agreed that the participant did not enjoy the task and removed the entry from the data. Additionally, the Cronbach's alpha for the creativity judges was $\alpha = 0.40$, indicating a poor agreement among them. We excluded the marks of one judge to raise the coefficient to $\alpha = 0.48$, that still indicates a low agreement but is somehow more acceptable. We provide additional details in the Limitations of the Study section.

For the Problem Solving task, one participant forgot complete the survey and was excluded from the data.

5.1.5.3 Hypothesis Testing The first three hypothesis regarding a difference in the best creativity score, the average creativity scores, and the number

of generated captions of the two groups are evaluated using t-tests. The assumptions of the t-test are all met and verified for all the three cases.

We summarize the results in Table 1 for the best creativity score, in Table 2 for the average creativity score, and in Table 3 for the number of generated captions.

H1 ₀	Factor	Mean diff.	DF	t-value	p-value
$\mu_{BCreat_{N-POS}} = \mu_{BCreat_{POS}}$	N-POS vs. POS	0.0488	39	-0.2060	0.8379

Table 1: t-test for $\mu_{BCreat_{N-POS}}$ and $\mu_{BCreat_{POS}}$

H2 ₀	Factor	Mean diff.	DF	t-value	p-value
$\mu_{Creat_{N-POS}} = \mu_{Creat_{POS}}$	N-POS vs. POS	0.0512	39	-0.3107	0.7577

Table 2: t-test for $\mu_{Creat_{N-POS}}$ and $\mu_{Creat_{POS}}$

H3 ₀	Factor	Mean diff.	DF	t-value	p-value
$\mu_{num.captions_{N-POS}} = \mu_{num.captions_{POS}}$	N-POS vs. POS	-1.2047	39	-1.2983	0.2018

Table 3: t-test for $\mu_{num.captions_{N-POS}}$ and $\mu_{num.captions_{POS}}$

For all the three hypothesis H1, H2, and H3, there are no significant differences. We do not reject the three null hypotheses and conclude that there is no evidence of a difference in creativity of software developers with respect to their pre-existing affective states.

We evaluate hypothesis H4, regarding differences in problem-solving skills for software developers with respect to the pre-existing affective states, with a t-test (unpaired, two-tailed). The difference of means for N-POS and POS is -0.06. There are significant differences between the variance of the two groups (F-test for variances, p-value = 0.0127). Therefore, we applied a t-test with Welch’s correction. The results of the test are summarized in Table 4.

With a p-value = 0.0079, we reject H4₀ in favor of H4₁. There are significant evidences in the differences of problem solving skills of computer science students with respect to their pre-existing affective states. A two-sample permutation test confirms the results, p-value = 0.0097.

H4 ₀	Factor	Mean diff.	DF	t-value	p-value
$ProbSolv_{N-POS} = ProbSolv_{POS}$	N-POS vs. POS	-0.0628	33.452	2.8212	0.0079

Table 4: t-test for $ProbSolv_{N-POS}$ and $ProbSolv_{POS}$

5.1.6 Discussion

In the previous section we wrote about the analysis performed against the retrieved data, and the results of the hypothesis testing.

In this section, we discuss the obtained results and compare them to the related studies. After that, we provide detailed information of the limitations of this study and how we tried to obviate to most of them. The last sub-section is dedicated to the *Lessons Learned* and suggestions for future researches.

The results of this study provide new elements in the controversial debate of pre-existing affective states and creativity. Additionally, the results provide statistical evidences that positive pre-existing positive affective states are indicators of higher analytical problem-solving skills.

We could not reject the null hypotheses that there is no difference in the creativity for N-POS and POS individuals. Our results agree with those of Sowden et al. (2011), that did not find differences between the positive and negative groups in terms of quality. Our results regarding the number of produced ideas are similar with those of the study, but not significant.

Our results deviate from those of Forgeard (2011), where non-depressed participants provided more creative captions. Nevertheless, it must be noted that we did not control the depression factor.

Additionally, we note in Figure 7 that there may be a tendency of higher creativity when the SPANE-B approaches extreme values such as -24 or 24. That is, it may be that higher creativity is achieved with both extremely positive and extremely negative affective states. The same holds for the average Creativity score, where the tendency could even be more stronger for the POS group (Figure 8). Unfortunately, the number of participants does not enforce the envision of particular relationships.

We found significant evidences that the participants in the POS group have a higher problem solving score with respect to the participants in the N-POS group. That is, there are significant evidences that software developers with the most positive pre-existing affective states also have the highest analytical problem-solving skills. We note in Figure 12 that there may be further interesting results if the whole scale of the SPANE-B is covered. We can not exclude that extremely negative pre-existing affective states could enhance problem solving thinking. The relationship between ProbSolv and SPANE-B appears to be linear and negative for the N-POS group. Furthermore, it is interesting to observe Figure 12 under another perspective. The points look to positively linearly increase right around the value 0 for SPANE-B, on both sides. This could possibly mean that both extremely positive and extremely negative pre-existing affective states are indicators of higher problem-solving capacities. We could only pro-

vide evidences for positive pre-existing affective states. A higher number of participants would provide further insights.

5.1.6.1 Limitations of the Study *Conclusion validity* concerns issues that affect the ability of drawing a correct conclusion. Although the number of the participants was acceptable, we stated in sections 5.1.5.1 and section 5.1.6 that we could not obtain a full coverage of the SPANE-B range in the negative direction. This limits the conclusions of the study, as participants with extremely negative affective-states could not be studied. Our study collected participants of a very specific population (Computer Science students) that was diverse in nationality but limited in the age of the participant (21.50 years old with $SD=3.01$ years) and year of study (2.26 years with $SD=1.38$ years). It is important to keep this limitation in mind when we generalize to the population of software developers.

The 42 participants tended to be positive in their affective states. That is, we were not able to collect participants extremely negative in their affective states. Even if using median splits, this study provides comparisons between what the literature would consider positive people and neutral people.

In our study, we used proper statistical tests. In particular, we carefully observed the assumptions of the t-test. For both the tasks, we checked the normality of the N-POS and POS groups using the Shapiro-Wilk test (p-value=0.4280 and 0.9127 for the creativity task, p-value=0.3964 and 0.1353 for the problem solving task). Additionally, we run a F-test of equality of variances for the two groups (p-value = 0.9868 for the creativity task, p-value = 0.0127 for the problem solving task) that brought us to adopt the Welch's t-test for the problem solving task. Furthermore, a non-parametric permutation test confirmed the rejection of the null hypothesis for the problem solving task, yielding a very similar p-value of the one obtained with the t-test. When we rejected the null hypothesis, the p-value was less than 0.01. This gives strength to the achieved results.

Our problem solving task is a mono-operation because all the participants played the same version of the Tower of London game. Besides this, we do not signal interaction of testing and treatment, hypothesis guessing or evaluation apprehensions.

We would like to bring to the reader's attention that the Cronbach's alpha for the judges was $\alpha = 0.48$. Thus, there was a low agreement between the judges for the creativity scores among the 220 captions. This may not imply particular implications to the study because of the different definitions that people give to creativity.

Internal validity threats are influences that can affect the independent variable with respect to causality, without the researcher knowledge.

The absence of knowledge of the participants about the existence of groups mitigated multiple-groups threats. The absence of knowledge about the experiment contents mitigated social threats to internal validity. Regarding maturation effects, we carefully observed the data and removed an entry about a

participant that looked bored with respect to the task. The participants happily collaborated and enjoyed the two tasks. This observation was successfully reflected by the two SPANE-B scores of the participants. The instrumentation and the data collection process were mostly automatized. The only risky part was the transcription of the captions in the spreadsheet that we distributed to the judges. We asked to third persons to check the transcriptions. No errors were detected.

Construct validity refers to whether the measurements model independent and dependent variables from which the hypothesized theory is constructed or not. As we explained in section 5.1.4.2, all the variables and the measurement instruments come from a literature review of the fields of software engineering, Psychology, and Cognitive Science. The creativity scores derives from experiments already used in previous studies. The problem-solving score derives from reviewing the literature. Although this metric was not formally defined elsewhere, it refers to analytical problem-solving tasks, where the results are unique, mathematically defined and calculated. The affective states variable comes from a measurement instrument used in Psychology. The hypotheses formulations directly derive from well-known research problems in Psychology and Cognitive Science.

5.1.6.2 Lessons Learned We describe this experiment as a positive experience, in which the reality adhered to the plans. The most important lesson learned is about the distribution of the affective states. Even if we were able to collect 42 participants, the SPANE-B score has never fallen below the value of -9. We expected a more homogeneous distribution of the participants with respect to the SPANE-B score. For future researches, we suggest either to adopt mood induction techniques or to gather a much higher number of participants. We believe that the second strategy requires less effort because this study is about pre-existing affective states, where the emotions are measured using the last 4 weeks as time reference.

We learned that the creativity tasks generates a high amount of documents to be collected (precisely $2n$) and each participant writes between 5 and 6 captions per picture. We encourage future studies to develop an electronic version of the task to minimize the effort needed and the errors that can occur when reporting them to the judges.

The Problem Solving task, namely the *Shallice test*⁴, resulted to be easy to be completed. Recalling from section 5.1.4.2 that the ProbSolv score is defined as the ratio between TOLSS and PTS, and that TOLSS component ranges between 0 and 36, TOLSS score average was 31 (SD=4.05). A more difficult version of the Tower of London game may lead to more variegation of the results.

⁴[1,2,3] pile heights, 3 disks, Shallice's 12 problems

5.1.7 Primary Empirical Conclusions

In this study we described a controlled experiment that provides empirical grounds about the role of pre-existing affective states for two important skills for software development. We provided detailed instructions on how to replicate the experiment and advices for future practitioners. What follows are the Primary Empirical Conclusions of this study, that will be further addressed in the Conclusion section of the thesis.

PEC 1 There are no significant evidences that developers with the most positive pre-existing states are more creative than developers feeling non-positive pre-existing states. While this may feed doubts about the existence of a relationship between affective states and creativity, we observed a weak trend of higher creativity with extreme affective states. This interpretation of our data may reconcile the positive and negative theories about mood and creativity. More researches with a higher number of participant are needed and encouraged.

PEC 2 There are significant evidences that developers feeling the most positive pre-existing states possess higher problem-solving skills. Moreover, we observed that there may be a V-shaped relationship between pre-existing affective states and problem-solving skills, symmetric with respect to the value 0 for the SPANE-B score. Future studies should involve a higher number of participants to further investigate this concern.

The implications for this study are meaningful for the IT industry, software engineering, and the educational branch of software engineering. As software developers possess higher problem-solving skills when feeling positive affective states in the long-run, future studies should address how to raise the general happiness of software developers (and Computer Science students). Additionally, we believe that it is necessary to study the affective states of software developers during a software development life-cycle.

We will recall our two Primary Empirical Conclusions in the Conclusion section of the thesis.

5.2 Study 2: Affective States and Software Development Productivity

Abstract

Background: Software engineering lacks studies about the role of affective states for Software Developer productivity, while other disciplines provided strong evidences of such role.

Objective: Provide a quantitative evaluation of the correlation of 3 mood dimensions (valence, arousal, and dominance) to the productivity of people developing software.

Method: Controlled, randomized, repeated measurements study with 8 participants. Four B.Sc students in Computer Science and 4 professional software developers working on their software projects (N=72 measurements). Fit of a linear mixed effects model to describe the general impact of the mood variables.

Results: Evidences that the valence mood dimension is positively related to productivity of software developers ($p=0.0000$). No evidences that the arousal mood dimension is correlated to the productivity. Evidences that the dominance mood dimension has a positive correlation with the productivity ($p=0.0292$). The model explains 30% of the deviance for the productivity.

Limitations: Productivity is self-assessed by the participants. Limited number of participants. Complete independence of the results because of the different software projects.

Conclusions: The results provided evidences of positive correlations of two mood dimensions (valence, dominance), one of which is a new entry in these types of studies (dominance). One mood dimension (arousal) is not supported to have a correlation with productivity but the related studies found evidences in favor of a positive correlation. This study is a first step to estimate the productivity of software developers by measuring emotions.

5.2.1 Introduction

In the Introduction section of the thesis we reported the need to study the role of software developer’s affective states. In particular, the productivity of individuals based on their affective states is a well-established subject of study in other disciplines such as Psychology. Software engineering studies ignore the mood of developers as a factor that influences productivity.

In this study we write about a study which implements Experience Sampling Method and repeated measures, within-subjects longitudinal data designs. We study the role that three dimensions of real-time affective states (valence, arousal, dominance) have on the productivity of software developers. The instruments that we use to measure the mood dimensions are supported by Psychology studies.

We propose a linear mixed effects model, which takes into account the dependency between the participant’s repeated measurements and the dependency that these repeated measurements have over time. Our model is then tested with 8 participants (4 B.Sc. Computer Science students and 4 professional software developers) who develop their specific software projects.

Our study presents first estimations of how the affective states are correlated with the productivity of programmers and may lead to future measurements of developers productivity basically by asking them how they feel about their work.

5.2.1.1 Motivation The purpose of this study is to provide an initial model for the impact of software developer’s affective states on their productivity, while programming. The motivation for this study are already stated in section 1.1. Therefore, we recall here the highlights for this study.

Even if software engineering lacks studies about the topic, the Psychology research established meaningful results since more than 70 years. We merge the two research disciplines in order to provide an explanation of the productivity of software developers while measuring their mood.

Through the use of repeated measurements, longitudinal, within-subjects analysis and linear mixed models, we show that both the mood and the productivity vary in short intervals of time. We claim that the mood is significant and positively correlated with the productivity of the developers.

To this end, we conducted a repeated measures study with 8 participants. The participants developed software for 90 minutes. We assessed their mood and their productivity every 10 minutes, using a Psychology related survey.

In the section 1.3, we stated the main goal of our thesis. Our work provides the first answers to the sub-question 3.

3. How do real-time affective states influence the immediate productivity of software developers?

We transform this research question in research objectives and research hypothesis in the *Goal Definition* section.

The implications of this study are relevant for the industry and could lead to measurements of the productivity of software developers via their emotions.

5.2.2 Related Work

In the previous section we stated the motivations of our research and introduced our research questions. While in software engineering there are no studies providing insights to our research questions, there is a body of knowledge in Psychology about affective states and productivity. We reported the related work about productivity in software engineering in section 2.1.1 of the thesis. We wrote about the related work on productivity and affective states in section 2.2.2 of the thesis.

Thus, in this section we report the related studies that provide empirical evidences. We will compare our results with these results in section 5.2.6 (Discussion).

There is little research on the role of emotions of software developers. We were able to identify two papers. The work of Shaw (2004) is a “research-in-progress” paper. It provides an overview of some existing frameworks and methodologies to measure emotions. The paper does not report about empirical experiments to answer specific research questions. Nevertheless, it observes that in 48 hours of interval, there are drastic changes in the mood of software developers. The work does not draw conclusions, but raises the need to research in the topic (Shaw, 2004).

The study by Khan et al. (2010) further develops and reports two empirical studies. The studies are related to the developer’s debugging performance with respect to their affective states. The term “performance” is a synonym of productivity in this study. We justify the choice of the term as productivity is still not clearly defined in software development. The results show that when the mood dimension of valence is kept high and the arousal dimension varies, there is a positive correlation with the debugging performance. This study requests for more research in the topic (Khan et al., 2010).

Because of the tiny body of knowledge in Computer Science, we add to this section a paper from the Psychology field. The work by Fisher and Noble (2004) is an Experience Sampling Method (ESM) study about real-time correlates of performance and emotions while working. The results of this study shows that there is a strong positive correlation between positive emotions and task performance, while there is a strong negative correlation between negative emotions and task performance. This study encourages further research about real-time performance and emotions (Fisher and Noble, 2004).

Our work makes another step forward and merges the two previously reported studies. This paper studies the variations of mood and productivity of software developers while programming. It assesses the correlation with productivity of mood related factors. Our study proposes a first model, which could lead to measurements of productivity by using mood dimensions.

5.2.3 Description of the Experiment

Our study is a within-subjects, ESM-like repeated measurements design. It uses a linear mixed effects model to explain the variation and the correlation of the

variables.

In this section we describe our experiment and the model that we derive from the experience. First, we define the research goal that provides details to the research question that we stated in the Introduction section. In the *Experimental Design* sub-section, we derive three research hypothesis from the research goals, using the GQM framework. Then, we provide details about the involved variables. In particular, we provide definitions to the three mood dimensions. After that, we provide a theoretical background about mixed effects models, the participants selection, the required instrumentation, and the validity evaluation. The subsequent sub-section is *Execution*, in which we provide details about the task that the participants perform.

5.2.3.1 Goal Definition The **object of study** is software developers and their productivity while developing software.

The **purpose** of the experiment is to evaluate the individual productivity based on three different mood dimensions. The experiment provides insights in what we can expect in terms of productivity when controlling the mood of developers.

The **perspective** is from the point of view of the researchers. The researchers want to evaluate the correlation of the mood of software developers and their productivity. The main effect studied in the experiment is the productivity while programming on a software project.

One specific aspect is emphasized for the **quality focus**. We focus on productivity. The productivity score is self-assessed by the participants.

The **context** of the experiment is natural settings. B.Sc. students in Computer Science and professional software developers perform the tasks. The ability to generalize from this specific context is further elaborated in the threats to validity section.

The experiment addresses a real problem, i.e. the evaluation of software developers productivity while controlling the mood. The use of natural settings - like the working environment - as an experimental context provides good opportunities to replicate the experiment. The required material per participant is a tablet to be used by the researcher. We further discuss the materials needed in the Experimental Design section.

We use the goal-question-metric model (Basili and Weiss, 1984, Jedlitschka and Ciolkowski, 2008) to define the goal, here summarized.

O₁ ANALYZE THE PRODUCTIVITY
FOR THE PURPOSE OF EVALUATION
WITH RESPECT TO THE MOOD DIMENSIONS (VALENCE, AROUSAL,
DOMINANCE)
FROM THE POINT OF VIEW OF THE RESEARCHER
IN THE CONTEXT OF B.SC. STUDENTS AND PROFESSIONAL
PROGRAMMERS CODING THEIR SOFTWARE PROJECTS.

5.2.3.2 Experimental Design

5.2.3.3 Hypothesis Formulation Software developers are human beings and feel affective states. Their productivity is influenced by their mood.

We already reported our research hypotheses H5, H6, and H7 in section 3.2 of the thesis. In this sub-section we further develop them and state the alternative hypothesis and the measures needed.

5. Null hypothesis, $H5_0$: There is no correlation with the valence mood dimension and the productivity of software developers

$$H5_0 : \beta_{valence} = 0$$

Alternative Hypothesis, $H5_1 : \beta_{valence} \neq 0$

Measures needed: Software developers valence score and productivity

6. Null hypothesis, $H6_0$: There is no correlation with the arousal mood dimension and the productivity of software developers

$$H6_0 : \beta_{arousal} = 0$$

Alternative Hypothesis, $H6_1 : \beta_{arousal} \neq 0$

Measures needed: Software developers arousal score and productivity

7. Null hypothesis, $H7_0$: There is no correlation with the dominance mood dimension and the productivity of software developers

$$H7_0 : \beta_{dominance} = 0$$

Alternative Hypothesis, $H7_1 : \beta_{dominance} \neq 0$

Measures needed: Software developers dominance score and productivity

5.2.3.4 Variables The choice of the mood dimensions - valence, arousal, and dominance - derives from the review of related works (section 2.2.1). We recall that valence is the intrinsic attractiveness (or adverseness) of an event, object, or situation (Lewin, 1935). Arousal is the sensation of being mentally awake and reactive to stimuli, or a response to a difficult challenge where skills are acceptable. Dominance is the sensation by which the individual skill's levels are higher than the challenge level for a task (Csikszentmihalyi, 1997a).

These three dimensions describe differences in affective meanings among stimuli and are self-assessed by the participants with the Self-Assessment Manikin (SAM) questionnaire (Bradley, 1994). SAM consists of 3 non-verbal, pictorial items that directly measures valence, arousal, and dominance associated with a person's affective reaction to a stimulus. Figure 16 in Appendix F is a screen capture of the survey. Each point of the three items has an associated figure. For our scopes, the stimulus of SAM is the programming task. That is, participants express how they feel about their task.

Participants self-assess the productivity using a likert item. The last item of the survey in Figure 16 is the sentence "My productivity is ". The participants

end the sentence, choosing the suitable end in the set $\{very\ low, below\ average, average, above\ average, very\ high\}$.

The four variables of our survey range from 1 to 5. A value of 3 means “perfect balance” or “average” between the most negative (1) and the most positive value (5). For example, a value of 3 for the valence variable means “absence of attractiveness and absence of aversiveness”. A value of 5 for the dominance variable means “complete control of the task and the situation”. A value of 3 for the productivity variable means “average productivity”.

We obtained values for the variables role, experience with the programming language, experience with the task.

The role of the participants is either “*professional*” (PRO) or “*student*” (STU).

The classic productivity factors - experience with the programming language and the task - are obtained from the pre-task interview with the participants. The interviewer asks the participants to describe the project they are working on, the task that they are going to perform, their experience with the task, and their experience with the programming language. After these open questions, the interviewer asks the participants to describe their experience with the programming language and the experience with the task by picking one among the set $\{low, medium, high\}$.

5.2.3.5 Linear, Mixed Effects Models A linear mixed-effects model is a linear model that contains both fixed effects and random effects. The definition of a linear mixed effects model given by (Laird and Ware, 1982) expresses the n_i -dimensional response vector y_i , for the i th group as

$$y = X\beta + Zb + \epsilon,$$

$$b \sim N(0, \Sigma)$$

$$\epsilon \sim N(0, \sigma^2 I)$$

Where:

y is a vector observations with mean $E(y) = X\beta$,

β is a vector of fixed effects,

b is a vector of independent and identically distributed random effects with mean $E(b) = 0$ and variance-covariance matrix $var(b) = G$,

ϵ is a vector of independent and identically distributed random error terms with mean $E(\epsilon) = 0$ and variance $var(\epsilon) = 0$,

X and Z are matrices of regressors relating the observations y to β and b .

5.2.3.6 Participants Units Our study is a repeated measurements design. Repeated measurements do not require a high number of participants in order to obtain meaningful results. However, our aim is to test our proposed model. A

requirement for the recruitment of participants is to have variance in the skills. That is, to have a balance in the role of the participants, in their experiences with the programming language, and their experience with the task they face.

5.2.3.7 Objects and Instrumentation Besides the material required to log the interview answers, the only instrument required is a suitable device that implements the SAM and the Productivity item. We used a tablet device and developed a specific on-line survey, as we report in Figure 16, Appendix F.

5.2.3.8 Validity Evaluation In order to minimize the bias related to the knowledge of the experiment procedure, we only asked the users to participate to a study for a duration of maximum two hours, in which they would work on their current project while being observed.

5.2.3.9 Execution Each participant faces a pre-interview in which basic demographic data, information about the project, the tasks, and developer's skills are obtained.

For a period of 90 minutes, the participant works on the task. The researcher observes the behavior of the individual. Each 10 minutes, the developer completes the short survey on the tablet device. That is, valence, arousal, dominance, and productivity are self-rated for 9 times per participant.

After the completion of the working period, the researcher conducts a post-task interview.

We include the detailed instructions given to each participant in Appendix E. Please note that we wrote the instructions for the SAM survey by following the technical manual by Lang (1999).

5.2.4 Analysis

In the previous section we provided detailed information for the reader in order to understand and eventually replicate our experiment.

In this section we describe the performed analysis, in order to obtain meaningful results to our research goals and the research hypothesis. First, we provide descriptive statistics about the gathered participants. Then, we provide and discuss the generated time-series graphs for all the participants, in which we compare the reported productivity and the mood variables. In the *Hypothesis Testing & Model Estimation* sub-section, we state our linear mixed effects model for the productivity and provide detailed information on how we estimate the effects of the mood variables over the productivity. The techniques for obtaining the significance of the parameters are different with respect to classic Anova and regression. Therefore, we provide explanations on how we obtain significance of the effects.

5.2.4.1 Descriptive Statistics As a first step to analyze the data, we provide descriptive statistics.

Participants We collected 8 participants, for a total of 72 measurements. The mean for the age was 23.75 (SD=3.29). Seven of them were male and the remaining one a female. We managed to have 4 participants who are first year B.Sc. Computer Science students and 4 professional software developers.

Four participants were Computer Science students working on course-related projects. The remaining 4 participants were professional software developers, working on job-related projects.

We summarize in Table 5 the characteristics of the participants. It is interesting to notice that roles are not always related to the experience. The professional participant P2 reported a LOW experience with the programming language while the student participant P8 reported a HIGH experience in both the programming language and the task type.

In Table 6 we summarize the characteristics of the projects and the implemented task. We can see that there is a high variety of project types and tasks. Five participants programmed using C++, while 2 of them with Java and the remaining one with Python.

id	role	age	gender	progr. lang. experience	task experience
P1	PRO	25	M	HIG	HIG
P2	PRO	26	M	LOW	HIG
P3	PRO	28	M	HIG	HIG
P4	PRO	28	M	HIG	HIG
P5	STU	23	F	LOW	LOW
P6	STU	20	M	LOW	LOW
P7	STU	20	M	LOW	LOW
P8	STU	20	M	HIG	HIG

Table 5: Participants

id	project	task (implementation of)	progr. lang
P1	Data collection for hydrological defense	Module for data displaying	Java
P2	Research Data Collection & Analysis	Script to analyze data	Python
P3	HR Manager for a Private School	Retrieval and display of DB data	Java
P4	Process & Product Metrics Analyzer	Retrieval and sending of metrics	C++
P5	Music Editor	Conversion of music score to pictures	C++
P6	Code Editor	Analysis of Cyclomatic Complexity	C++
P7	CAD	Single-lined labels from a standard	C++
P8	SVG Image Editor	Multiple objects on a circle or ellipse	C++

Table 6: Projects and Tasks

Task Analysis All the participants completed the task and answered to all the questions. Therefore, we collected 72 measurements. We report in Table 7 the mean, the standard deviation, and the median for the variables.

	Productivity	Valence	Arousal	Dominance
Mean	2.63	3.04	3.08	3.15
Std. Dev.	0.94	0.92	0.98	1.11
Median	3	3	3	3

Table 7: Productivity, valence, arousal, and dominance mean, std. dev., and median

We provide in Figure 13, Figure 14, and Figure 15 the charts representing the changes of productivity over time with respect to each mood dimension. In all the three figures, the light-blue, solid line represents the productivity. The orange, dashed line represents a mood dimension.

There is no stable and shared metric for assessing the mood across persons. Nevertheless, it is sensible to assume that there is a reasonable metric within person. That is, a score of 1 of valence for a person may be equivalent to a score of 3 for another person. Therefore, as many Experience Sampling Method researches do, we transform the scores given by each participant to productivity and the mood dimensions to their respective Z-scores. That is, we express by how many standard deviations an observation is above or below the mean of the whole set of individual’s observations. In this way, we can have dimensionless, comparable measures between participants (Hektner et al., 2007).

As we can see in Figure 13, there are cases in which the valence score provides strong predictions of the productivity (participants P2, P7, and P8). For many intervals, e.g. P5 at interval 7, and P4 at intervals 4-7, the trends of both lines are perfectly matched. Participant P1 is the only one for which the valence does not provide strong predictions. Nevertheless, the valence Z-score is closely related to the productivity Z-score. There are very few cases in which the valence Z-score is more than a standard deviation distant than the productivity Z-score.

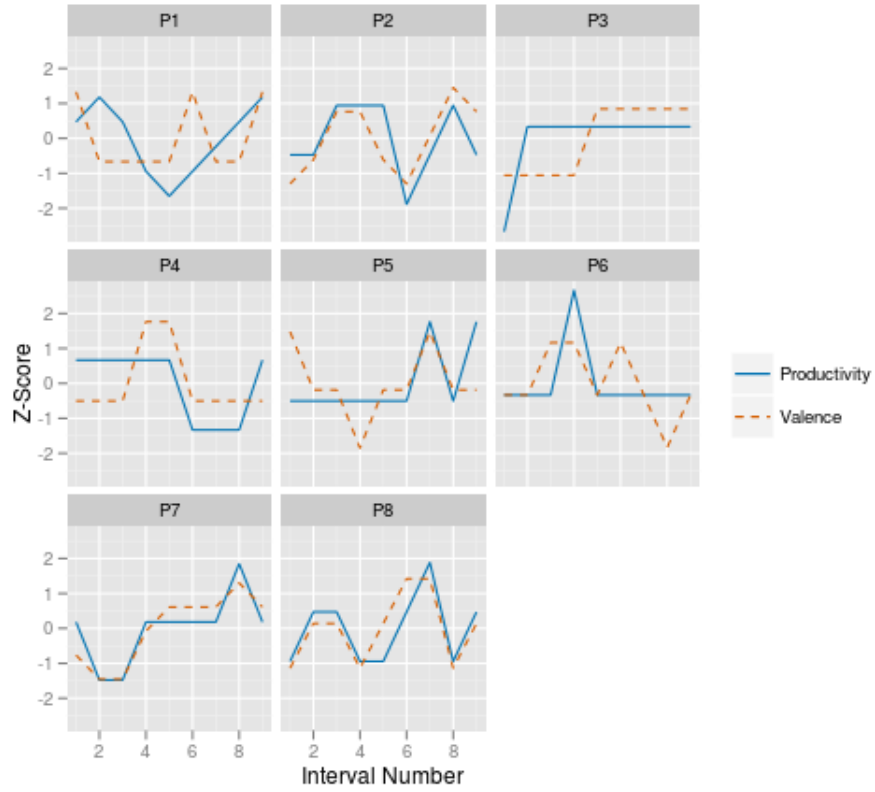


Figure 13: Productivity & Valence over Time

The arousal dimension, in Figure 14, seems less related to the productivity. Except from the cases P3, P4, and P7, the behavior of the arousal line often deviates from the trend of the productivity line. Nevertheless, there are intervals in which the arousal is still closely related to productivity, like with participant P8.

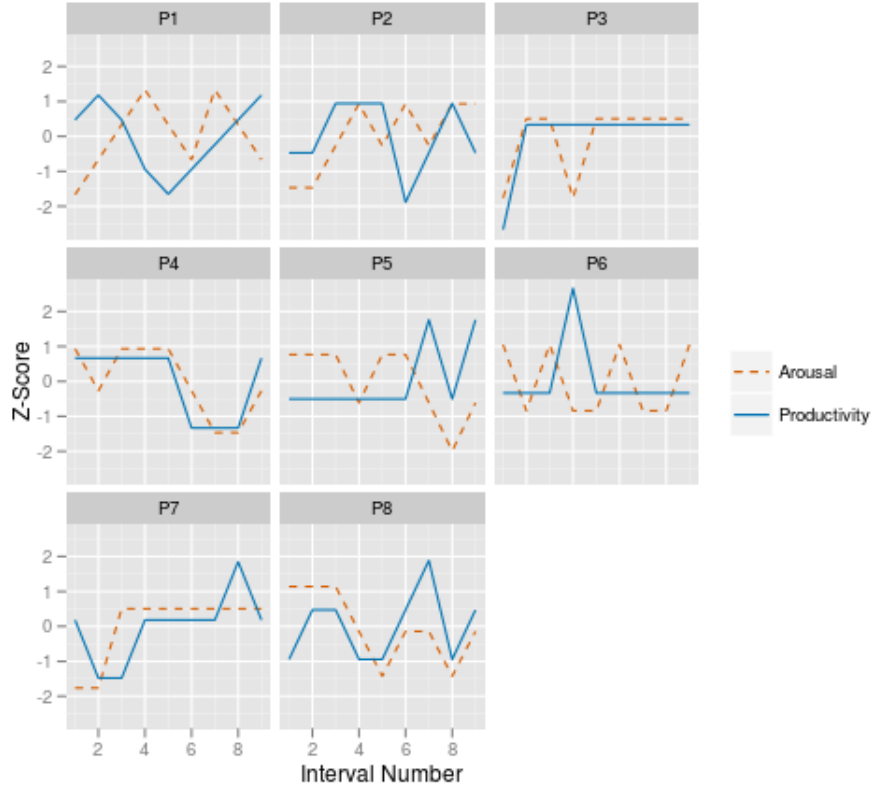


Figure 14: Productivity & Arousal over Time

In Figure 15 we plot the lines of productivity and the dominance dimension over time. The dominance dimension looks more correlated to productivity than the arousal dimension. Participants P1, P5, and P7 provided close trends. For the other cases, there are intervals in which the correlation looks close and stronger - e.g. P3 from interval 2 to interval 6 - but the correlation becomes weaker for the remaining intervals. For participant P6, it looks like there is not a clear correlation between dominance and productivity.

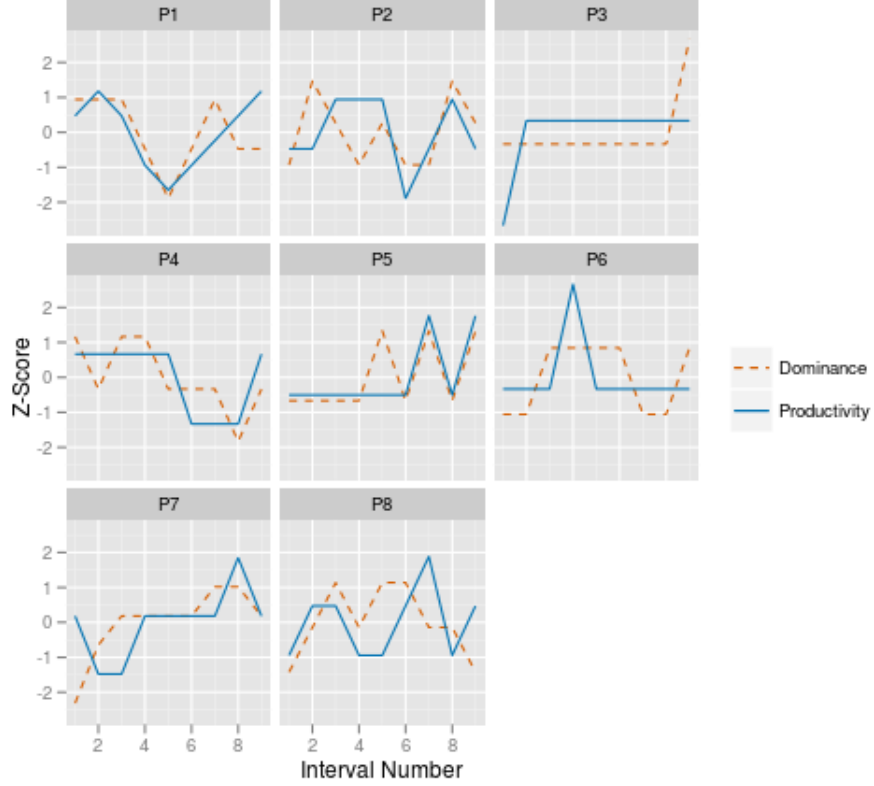


Figure 15: Productivity & Dominance over Time

The observation of the graphs already brings interesting insights. Because productivity is self-assessed by the participants, we believed that the dominance dimensions would have the most closely correlated measure. Instead, the valence dimension (that is, the happiness, the attractiveness, and the satisfaction related to the task) appears to be the one with the higher correlation.

We observe that for all the variables, there are changes approximately greater or equal than two standard deviations. In a small working time (90 minutes), there are strong variations of both the mood and the productivity. This is a first evidence that the classic measurements of the productivity provide central values and may not be suitable for representing whole situations. The same may hold for many productivity factors, which could be indicators of productivity for general, long-run assessments.

5.2.4.2 Hypothesis Testing & Model Estimation As we wrote in the *Description of the Experiment* section, our work adopts a mixed effects model. Our model estimates changes in the productivity Z-score through mood dimen-

sions. We implemented the model using R and the R package *lme4*⁵. We analyze the model using the R package *LMERConvenienceFunctions*⁶.

The estimation of the significance of the effects for mixed models is an open debate. Therefore, we now explain how the estimation is computed. The function *pamer.fnc* of the *LMERConvenienceFunctions* constructs an Anova table with upper-bound and lower-bound (anti-conservative and conservative, respectively) degrees of freedom, p-values, and deviance explained.

Upper-bound p-values are computed by using the number of data points minus the number of fixed effects. Lower-bound p-values are computed by using as denominator degrees of freedom the number of random effects multiplied by the number of participants. The amount of deviance explained by each fixed effect is the sum of squares of the effect, divided by the total sum of squares.

For the model construction, we have valence, arousal, and dominance as fixed effects, plus their interactions with time. The random effects is the time given the participant. That is, we express each measurement moment as a random effect, for each participant. In this way, we take into account the dependency of measurements within the participants, at the participant level and at a time level.

The full model is

$$productivity \sim (valence + arousal + dominance) * time + (time | participant),$$

where

productivity is the dependent variable; valence, arousal, dominance, and time are fixed effects; (time|participant) represents the random effect of time grouped by each participant.

The full model significantly differs from the null model ($productivity \sim 1 + (time | participant)$). We checked for normality and homogeneity by visual inspections of a plot of the residuals against the fitted values, plus a Shapiro-Wilk test.

In Table 8 we provide the parameter estimation for the fixed effects.

We have significant evidences to reject the first null hypothesis in favor of the alternative hypothesis. There are significant evidences of a (positive) correlation with the valence mood dimension and the productivity of software developers.

We do not have significant evidences to reject the second null hypothesis in favor of the alternative hypothesis. There are no significant evidences of a correlation with the arousal mood dimension and the productivity of software developers. This will further be discussed in the section, as it is the most surprising result.

We have significant evidences to reject the third null hypothesis in favor of the alternative hypothesis. There are significant evidences of a (positive) correlation with the dominance mood dimension and the productivity of software developers. We believed that the highest positive correlation with a self reported productivity would have been with the dominance variable. We further discuss

⁵<http://cran.r-project.org/web/packages/lme4/index.html>

⁶<http://cran.r-project.org/web/packages/LMERConvenienceFunctions/index.html>

this detail in the next section.

Fixed Effect	Value	Sum. Square	F-value	upper p-val (64 d.f.)	lower p-val (48 d.f.)	Dev. Ex- plain. (%)
valence	0.233*	16.870	25.255	0.0000	0.000	25.993
arousal	0.153	0.000	0.000	0.9937	0.9937	0.001
dominance	0.254*	3.3242	4.976	0.0292	0.0304	5.122
time	0.003	0.018	0.027	0.8683	0.8684	0.028
valence:time	0.040	0.279	0.418	0.5201	0.5209	0.430
arousal:time	-0.042	0.757	1.129	0.2918	0.2932	1.162
dominance:time	0.001	0.000	0.001	0.9712	0.9713	0.001

Table 8: Full-Model Coefficient Estimation, *p < 0.05

5.2.5 Discussion

In the previous section we wrote about the analysis performed against the retrieved data, the estimation of the mood effects against productivity, the results of the hypothesis testing and the significance of the results. In this section, we discuss our results and compare them with the related work. After the discussion of the results, we provide the most important limitations of our study.

Our study indicates that the three mood dimensions of valence, arousal, and dominance explain more than the 30% of the deviance of the self-reported productivity of software developers. In the previous section, we reported that we have significant evidences of a positive correlation of two mood dimensions (valence, dominance) and the productivity.

The most striking observation is that we do not have evidences for a positive correlation between arousal and productivity. Our results differ from those of Khan et al. (2010). Additionally, the study by Csikszentmihalyi (1997) classifies the mental state of the flow as the intersection between high arousal and high dominance. One possible explanation is that our limited number of participants did not understand the concept of arousal. All of them requested additional explanations for the SAM item related to arousal (Figure 16, Appendix F).

Our results about the valence dimension are in line with those of Khan et al. (2010) and Fisher and Noble (2004). We found significant evidences that valence is positively correlated with the productivity of software developers. Moreover, the valence effect alone accounts for the 25% of the total deviance of the productivity.

As far as we know, there are no studies which include dominance as a factor in productivity estimation. We found significant evidences that this mood dimension is positively correlated with the productivity. What surprises us is that dominance is not the main effect for the productivity. We were skeptical to include this mood dimension in the study, because we expected that the dominance values would have been too much close to a self-assessed productivity,

even coincident. Instead, the dominance effect estimated value is nearly identical of the valence estimated value (both slopes are near 0.25). Nevertheless, dominance accounted for the 5% of the productivity deviance, compared to the 25% explained by valence. This further encourages us to keep the dominance dimension in the next studies.

A possible explanation of no significant interactions between the mood dimensions and the time is that each participant worked on different, independent projects. This may change in future experiments, if a group of software developers simultaneously work on the same project.

5.2.5.1 Limitations of the Study Our study tests a model and provide initial, meaningful results. We have a limited number of participants (8) who provided a total of 72 measurements. Even if in repeated measurements designs a high number of participants is not required, we encourage a repetition of this study with a higher number of participants and a wide number of measurements per participant.

A working period of 90 minutes is interesting to evaluate the productivity in the short run. In fact, we showed that there are substantial variations of the productivity even in intervals of 10 minutes. Nevertheless, 90 minutes is a relatively short time.

The choice of a self-reported productivity is practical. The participants were selected at random and we could know about the programming language and their task only when they presented to the study. Even if we suspected a high positive - but misleading - correlation with the dominance variable and the perceived productivity, the results showed that this is not the case.

5.2.6 Primary Empirical Conclusions

In this study we described a linear mixed effects model to measure the immediate productivity of software developers over many, short intervals of time, using real-time, mood-related variables. We conducted a study to provide empirical results of the model. We showed that two dimensions of the mood (valence, dominance) have positive, significant correlation with the productivity. We found that one dimension of the mood (arousal) does not have a significant correlation with the productivity.

Here we report the three Primary Empirical Conclusions of this study, that will be recalled in the Conclusion section of the thesis.

PEC 3 There are significant evidences of a positive linear correlation between the valence dimension that software developers feel when programming, and the productivity of their task. For this mood dimension, our study is in line with the previous studies.

PEC 4 There are no significant evidences of a linear correlation between the arousal dimension that software developers feel when programming, and the

productivity of their task. Our study is not in line with the related works for the arousal dimension, but we believe that a possible explanation is due the tiny number of the participants and misunderstanding of the instructions.

PEC 5 There are significant evidences of a positive linear correlation between the dominance dimension that software developers feel when programming, and the productivity of their task. Our study introduces dominance as a new variable for the mood measurement, which has never been used for performance assessment.

As far as we know, this is the first attempt in Computer Science to determine the productivity of software developers by measuring their affective states. It is one of the first attempts to redefine the productivity by merging well-established, interdisciplinary proposals. Our results encourage future studies in this field. Software development comprises activity strongly influenced by creativity and problem solving skills. We showed in the first part that the affective states have a role in this. In this section, we showed that the real-time mood measurements of software developers have predicting powers over their productivity.

Our model needs future development. Future studies should provide results using people working on the same software project and compare acceptable productivity metrics with their mood dimensions. With acceptable productivity metrics, we mean classic ratio factors or non-generalizable metrics (e.g., metrics internally defined and adopted by a company).

The implications of our study are both theoretical and practical. The whole IT industry and empirical software engineering should further research in this field. If our results are replicated and consolidated by future experiments, the Industry world could begin to measure the productivity of developers considering them as human beings. Additionally, future studies may find ways to enhance the mood dimensions of software developers in order to improve their productivity. Our results indicate that the mood associated with the programming task stimulus is positively correlated with its associated productivity. Therefore, Industry and software engineering research may lead to ways to make developers perceive positive emotions about their own work in order to boost their related productivity.

6 Discussion

The two studies described in section 5 provide empirical research about the role of affective states for software developers. In this section, we expand the Primary Empirical Conclusions of the studies and discuss their theoretical and practical implications.

Study 1 investigates about roles of affective states of software developers and important skills tighten to convergent and divergent thinking, namely analytical problem-solving and creativity.

From our Primary Empirical Conclusion (PEC) 1, obtained from researching hypotheses H1, H2, and H3, we concluded that there are no significant evidences that developers with the most positive pre-existing states are more creative than developers feeling non-positive pre-existing states. However, the arrangements of the points in the graphs suggest that this correlation may still exist, but more studies are needed with a higher number of participants. Additionally, the judges experts in the field of creativity come from different domains (creative writing, design & arts, cinema) and have disagreements about the concept of creativity. Our results agree with those of Sowden and Dawson (2001), who do not report significant differences between affective states and the quality of the creative work. Our results also agree about the number of generated ideas, although non significantly. Our results are in contradiction with those of Forgeard (2001), where participants low in depression performed significantly better in creativity generation tasks under negative mood. However, it must be noted that our study does not measure the depression levels. Our data suggests relationships described in the meta analysis of Davis (2009), where the literature agrees on no significant differences between positive and negative moods.

From PEC 2, obtained by investigating hypothesis H4, we found significant evidences that developers feeling the most positive pre-existing states possess higher analytical problem-solving skills. Moreover, the patterns that emerge by observing the data suggest a V-shaped relationship between the problem-solving score and the SPANE-B affect balance score. More studies are needed to further investigate the raised issue and to support our findings. Our results agree with those of Kaufmann and Vosburg (1997), where participants feeling positive affective states performed non-significantly better in problem-solving tasks. Our results are significant and support their hypotheses. Our results are in contradiction with those of Abele-Brehm (1992) and Melton (1995), where the results show higher problem solving skills for participants feeling negative affective states. Additionally, our results are in line with those of Brand et al. (2007), who report lower cognitive processing abilities under negative emotions. Our study reports new results obtained from the observation of the data. When using the SPANE measurements instrument for affective states, we note that there may be a V-shaped relationship of the analytical problem-solving skills, having a vertex in proximity of the value 0 for the SPANE-B score. Future studies with a higher amount of participants should address this observation. Additionally, we noted the difficulty to randomly find participants feeling strong negative affective states. Future studies should address this issue, by either using

a much higher number of participants or adopt mood induction techniques in the long run.

As far as we know, Study 1 is a novelty in the field of software engineering. Future research should study important skills for software development and the impact that long-run affective states provide on such skills.

Study 2 is a proposal to measure the productivity of software developers as individuals, through their emotions about the task they are performing. To this end, we defined a linear mixed effects model that permits experiments with repeated, longitudinal data. Through a first study, we found that the model was capable to describe the 30% of the productivity deviance. This study is a novelty for software engineering. Therefore, our comparisons with the related work is either from other disciplines or not strictly related to the same affective states variables.

From PEC 3, obtained by investigating hypothesis H5, we found significant evidences for a positive correlation between the valence mood dimension and the productivity of software developers.

From PEC 4, obtained by investigating hypothesis H6, we could not find significant evidence for a positive correlation between the arousal mood dimension and the productivity of software developers. This is in contrast from the literature review. However, the tiny number of participants, the high dispersion of their personal data tendencies on arousal, and the number of questions raised about arousal in the pre-task interview, make us suspect that they could not really understand this variable.

From PEC 5, obtained by investigating hypothesis H7, we found significant evidences for a positive correlation between the dominance mood dimension and the productivity of software developers. Although dominance is well-known in Psychology, our study introduces it as a new variable for estimating the productivity. Recalling that the dominance dimension reflects the perception of the adequacy of the participant skills with respect to the task being performed, we expected the dominance to be the variable with the highest explanation powers about productivity. However, it could only explain the deviance of productivity 5 times less than the valence dimension. This further indicates that the general “happiness” about the task being performed has more explanation power on the productivity than the sensation of adequacy of our skills.

Our results are in line with the results of Khan et al. (2010), where a high valence resulted in the best performance while debugging software. However, our results are not in line with this study regarding the arousal dimension. We have already reported a possible explanation for this. Our results are also in line with the study of Fisher and Noble (2004), where positive affective states are found to be positively correlated with productivity, and negative affective states are found to be negatively correlated with task productivity. Additionally, our results agree at a theoretical level with the studies of Csikszentmihalyi (1997a), where the concept of flow (high arousal and high dominance) is linked with the productivity of a task. The previous explanations for the arousal dimension hold here.

Study 2 provides completely new results for the field of software engineering,

where the productivity of software developers is measured through their affective states on their task. We showed that both the productivity and the dimensions of valence, arousal, and dominance are dynamic for software developers and fluctuate many times in a relatively short interval of 90 minutes. This should be kept in mind even in future studies about productivity that will not take into account affective states of the software developers.

6.1 Theoretical and Practical Implications

We believe that this thesis provides a step forward in the research of psychometrics in software engineering and the consideration of software developers as unique individuals.

The theoretical implications of PEC 1 and PEC 2 are that pre-existing, positive affective states are indicators of higher analytical problem-solving skills. Although a correlation does not imply a causation, our findings are also practical: the measurement of pre-existing affective states can be used to assess problem-solving skills for software developers. More studies are needed to see if the same holds for creativity.

The theoretical implications of PEC 3, PEC 4, and PEC 5 are that positive, real-time affective states of a development task are correlated with the programmer's productivity. More specifically, the valence and the dominance mood dimensions have a positive linear correlation with the self-assessed productivity. Additionally, a positive valence indicates attractiveness and happiness with the task being performed. A high happiness with the task has 5 times more explanation power of the productivity of the task than the sensation of having adequate skills (dominance). In other words, how much developers like their work is an indicator of how much they are productive. The measurement of software developers productivity through their affective states explains the 30% of the deviance of the productivity. While this might not be sufficient to only rely to affective states for productivity measurement, affective states should be considered a productivity factor of non-trivial importance. The literature suggest that arousal plays the same role with productivity, but we could not find significant evidence. The practical implications of our findings may lead to measurement of the productivity either by only measuring affective states of IT firms workers or to combine internal, well-established productivity metrics with the affective states component. Additional evidences of our findings may support positive mood-induction techniques in order to boost the productivity of software developers.

7 Conclusions

In this thesis we reported two empirical studies, which provide initial insights to the role that emotions have in software engineering. In particular, we stated a general research question in the thesis Introduction section. This research

question will not completely be answered in the next years. Nevertheless, we provided initial steps to lay down the research in the field.

In the Introduction section we stated the motivation for our research, the research problem, and the questions that this thesis aims to answer.

From the related work (section 2), we discovered that software engineering tries to define productivity in software development since about 40 years but the problem is still open. Nevertheless, we noted that every software developer is able to express feelings on the productivity and to quantify it in a general way. The related work highlighted the need to study software developers as human beings, and to adopt psychometrics in empirical studies. Psychology and Cognitive Science provide a huge body of knowledge about the role of affective states on skills and productivity. Software engineering lacks such studies for software development. We also reported that there is a call for research in software engineering using psychometrics, tentatively called Individualized Software Engineering (ISE). This study further lays down the foundations for ISE and calls for future research in this field.

We built our research framework (section 3) from the information gathered from the related work. The research framework was employed together with the research goals to construct the research hypotheses that this thesis aims to address.

In the Empirical Research Design section (number 4), we provided a brief review of the empirical investigations that software engineering adopts in order to research hypotheses. We reported that our experiments are controlled experiments, and we introduced the concept of Experience Sampling Method and linear mixed models.

In section 5, we reported our two studies, written as almost self-contained research papers. The two studies, namely *Study 1: Affective States and Software Development Skills* and *Study 2: Affective States and Software Development Productivity*, provide answers to the research questions formulated in section 1.3 and empirical evidences to either support or reject the hypotheses reported in section 3.2.

In the Discussion section of the thesis (section 6), we expanded the Primary Empirical Conclusions of the two studies and provided theoretical and practical implications of our findings.

In the two remaining sub-sections, we provide our answers to the research questions of the thesis. Then, we suggest future research opportunities for Individualized Software Engineering.

7.1 Answers to the Research Questions

In section 1.3 of the thesis we reported the research questions of this study. The aim of this work is to provide answers to the research question “How do the Affective States impact on Skills and Productivity in Software Development?”. We broke down the research question in three sub-questions to be answered by our studies. We re-write here the sub-questions:

1. How do pre-existing affective states influence the creativity of software developers?
2. How do pre-existing affective states influence the analytical problem-solving skill of software developers?
3. How do real-time affective states influence the immediate productivity of software developers?

Our Primary Empirical Conclusions contain answers for these research questions.

In PEC 1, we stated that we could not find significant evidences that developers with the most positive pre-existing affective states are more creative than developers feeling non-positive pre-existing states. However, we noted that the trends of the data indicate a weak correlation that future studies should verify, with a higher number of participants. This correlation would indicate a positively correlation with extremely positive and extremely negative affective states. The body of knowledge in Psychology and Cognitive Science reports that affective states indeed influence the creativity of individuals. However, as the body of knowledge in other disciplines reports, a more definite answer can be provided with future research.

We reported in PEC 2 that there are significant differences in analytical problem-solving skills of software developers, with respect to their pre-existing affective states. More specifically, this difference is translated as a higher problem-solving capacity for software developers feeling the most positive affective states. While we can not yet speak about causation, as it is outside the scopes of this work, we can provide a clear answer to sub-question number 2.

In PEC 3 and in PEC 5, we stated that there are significant evidences of a positive, linear correlation with the affective states dimensions of valence and dominance with the productivity of software developers. Even if in PEC 4 we could not find a correlation with productivity and the arousal dimension, the literature suggests that this correlation exists. Again, causation is outside the scopes of our study, but we can answer to sub-question number 3.

7.2 Future Research Opportunities

Future researchers should provide additional details to our claims and may study how positive inductions to the mood of programmers enhances their productivity. The results of our study suggest that how software developers perceive emotions about their tasks is strictly related to their effective productivity.

Regarding pre-existing affective states and software development skills, we have suggestions for future research. There is the need to study about how to raise positive affective states of developers and assess their skills. Long-term researches should study affective states of software developers from the hiring phase. In particular, there is the need to see how software developers' personalities and affective states are correlated with their work-related achievements and careers and how this translates to an improvement of the quality of software.

Additionally, there is need to research how mood-induction techniques enhance important skills in the long-term. Moreover, software engineering for education should assess how raising positive affective states of Computer Science students may higher their results in assessments.

Regarding the productivity of software developers, we suggest to perform case studies in IT companies to compare the affective states of engineers and programmers with internal, established metrics of productivity. If more evidences support our findings, we envision the creation of feedback systems in which the productivity of a whole IT firm is measured with affective states measured at the personal level, at the teams level, at the departments level, and at the customers level.

As far as we know, this is the first study in Computer Science that proposes an estimation of the productivity through deep human factors such as the emotions of system developers. It is another step forward to give credit to the humanity side of the developers, who program using creativity and analytical problem-solving skills. Additionally, this thesis provides evidence that the field of ISE should be expanded and taken into consideration as one of the major fields of software engineering.

Software developers are human beings, and as unique individuals they should be considered when developing software processes, development tools, and working environments.

References

- A. Abele-Brehm. Positive versus negative mood influences on problem solving: A review. *Polish Psychological Bulletin*, 23:203–221, 1992.
- P. Abrahamsson. Commitment development in software process improvement: critical misconceptions. *Proceedings of the 23rd International Conference on Software Engineering ICSE 2001*, pages 71–80, 2001. doi: 10.1109/ICSE.2001.919082.
- A. Albrecht. Measuring Application Development Productivity. In I B M Press, editor, *Proceedings of the Joint SHAREGUIDEIBM Application Development Symposium*, volume 83, pages 83–92. IBM, IBM Cooperation, 1979.
- A. Albrecht and J. Gaffney. Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation. *IEEE Transactions on Software Engineering*, SE-9(6):639–648, November 1983. ISSN 0098-5589. doi: 10.1109/TSE.1983.235271.
- M. Baas, C. De Dreu, and B. Nijstad. A meta-analysis of 25 years of mood-creativity research: hedonic tone, activation, or regulatory focus? *Psychological Bulletin*, 134(6):779–806, 2008.
- V. Basili and D. Weiss. A methodology for collecting valid software engineering data. *IEEE Transactions on Software Engineering*, SE-10(6):728–738, 1984.
- K. Beck. *Extreme Programming Explained: Embrace Change*. The XP Series. Addison-Wesley, 1999. ISBN 0201616416.
- K. Beck, M. Beedle, A. Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland, and D. Thomas. Manifesto for Agile Software Development, 2001.
- B. Boehm. Understanding and Controlling Software Costs. *IEEE Transactions on Software Engineering*, 14:1462–1477, 1990.
- L. Bradley. Measuring emotion: the self-assessment semantic differential. *Journal of Behavior Therapy and Experimental Psychiatry*, 25(I), 1994.
- S. Brand, T. Reimer, and K. Opwis. How do we learn in a negative mood? Effects of a negative mood on transfer and learning. *Learning and Instruction*, 17(1):1–16, February 2007. ISSN 0959-4752. doi: 10.1016/j.learninstruc.2006.11.002.
- E. Chen. Program complexity and programmer productivity. *Software Engineering, IEEE Transactions on*, SE-4(3):187–194, May 1978. ISSN 0098-5589. doi: 10.1109/TSE.1978.231497.

- E. Chrysler. Some basic determinants of computer programming productivity. *Communications of the ACM*, 21(6):472–483, June 1978. ISSN 00010782. doi: 10.1145/359511.359523.
- A. Cockburn and J. Highsmith. Agile software development, the people factor, 2001. ISSN 00189162.
- J. Collofello and S. Woodfield. Software productivity measurement. *Proceedings of the National Computer Conference, 1983*, pages 757–762, 1983.
- L. Cronbach. Coefficient alpha and the internal structure of tests. *Psychometrika*, 16(3):297–334, September 1951. ISSN 0033-3123. doi: 10.1007/BF02310555.
- M. Csikszentmihalyi. Finding flow. *Psychology Today*, 30(4):46, 1997a. ISSN 00333107.
- M. Csikszentmihalyi. *Creativity: Flow and the Psychology of Discovery and Invention*. Harper Perennial, 1997b. ISBN 0060928204.
- C. Dale and H. Van Der Zee. Software productivity metrics: who needs them? *Information and Software Technology*, 34(11):731–738, 1992. ISSN 09505849. doi: 10.1016/0950-5849(92)90168-O.
- D. Darcy, Meng J. Ma, and R. Smith. Exploring Individual Characteristics and Programming Performance : Implications for Programmer Selection. *Sciences-New York*, 14(12):1–10, 2005.
- M. Davis. Understanding the relationship between mood and creativity: A meta-analysis. *Organizational Behavior and Human Decision Processes*, 108(1):25–38, January 2009. ISSN 0749-5978. doi: 10.1016/j.obhdp.2008.04.001.
- E. Diener, D. Wirtz, W. Tov, C. Kim-Prieto, D. Choi, S. Oishi, and R. Biswas-Diener. New Well-being Measures: Short Scales to Assess Flourishing and Positive and Negative Feelings. *Social Indicators Research*, 97(2):143–156, May 2009. ISSN 0303-8300. doi: 10.1007/s11205-009-9493-y.
- L. Ding, Q. Yang, L. Sun, J. Tong, and Y. Wang. Evaluation of the capability of personal software process based on data envelopment analysis. In *International Software Process Workshop*, volume 3840 LNCS of *Lecture Notes in Computer Science*, pages 235–248. Springer Verlag, 2006.
- J. Dow. External and Internal Approaches to Emotion, 1992.
- R. Feldt, R. Torkar, L. Angelis, and M. Samuelsson. Towards individualized software engineering: empirical studies should collect psychometrics. In *Human Factors, CHASE '08*, pages 49–52. ACM, 2008. ISBN 9781605580395. doi: 10.1145/1370114.1370127.
- G. Fischer. Cognitive View of Reuse and Redesign. *IEEE Software*, 4(4):60–72, 1987. ISSN 07407459. doi: 10.1109/MS.1987.231065.

- C. D. Fisher and C. Noble. A Within-Person Examination of Correlates of Performance and Emotions While Working. *Human Performance*, 17(2):145–168, 2004. ISSN 08959285. doi: 10.1207/s15327043hup1702.
- M. J. Forgeard. Happy people thrive on adversity: Pre-existing mood moderates the effect of emotion inductions on creative thinking. *Personality and Individual Differences*, 51(8):904–909, December 2011. ISSN 0191-8869. doi: 10.1016/j.paid.2011.07.015.
- M. H. Halstead. *Elements of Software Science*, volume 7 of *Elsevier Computer Science Library. Operating And Programming Systems Series; 2*. Elsevier, 1977. ISBN 0444002057.
- J. M. Hektner, J. A. Schmidt, and M. Csikszentmihalyi. *Experience Sampling Method: Measuring the Quality of Everyday Life*. Sage Publications Inc, 2007. ISBN 1412949238.
- IEEE. *IEEE Standard for Software Productivity Metrics*, volume 1992. IEEE software Society, 1992. ISBN 1559372583.
- R. Ilies and T. Judge. Understanding the dynamic relationships among personality, mood, and job satisfaction: A field experience sampling study. *Organizational Behavior and Human Decision Processes*, 89(2):1119–1139, 2002. ISSN 07495978. doi: 10.1016/S0749-5978(02)00018-3.
- A. Jedlitschka and M. Ciolkowski. Reporting experiments in software engineering. *Guide to Advanced Empirical Software Engineering*, pages Section II, 201–228, 2008.
- V. Kadar. *On Application of Earned Value Index to Software Productivity Metrics in Embedded Computer Systems*. IEEE Comput. Soc. Press, 1992. ISBN 0818627603. doi: 10.1109/CMPEUR.1992.218455.
- J. C. Kaufman, J. Lee, J. Baer, and S. Lee. Captions, consistency, creativity, and the consensual assessment technique: New evidence of reliability. *Thinking Skills and Creativity*, 2(2):96–106, November 2007. ISSN 18711871. doi: 10.1016/j.tsc.2007.04.002.
- G. Kaufmann and J. Vosburg. 'Paradoxical' Mood Effects on Creative Problem-solving. *Cognition & Emotion*, 11(2):151–170, 1997. ISSN 02699931. doi: 10.1080/026999397379971.
- I. A. Khan, W. Brinkman, and R. M. Hierons. Do moods affect programmers' debug performance? *Cognition, Technology & Work*, 13(4):245–258, October 2010. ISSN 1435-5558. doi: 10.1007/s10111-010-0164-1.
- N. M. Laird and J. H. Ware. Random-Effects Models for Longitudinal Data. *Biometrics*, 38(4):963–974, 1982. ISSN 0006341X. doi: 10.2307/2529876.

- A. Langley. Strategies for Theorizing from Process Data. *Academy of Management Review*, 24(4):691, 1999. ISSN 03637425. doi: 10.2307/259349.
- R. Larson and M. Csikszentmihalyi. The experience sampling method. *New Directions for Methodology of Social and Behavioral Science*, 15(15):41–56, 1983. ISSN 14690292. doi: 10.1177/0894439305281844.
- K. Lewin. *A dynamic theory of personality*, volume 2008. McGraw-Hill, 1935. ISBN 9781443730334. doi: 10.1109/IEMBS.2008.4649489.
- R. J. Melton. The Role of Positive Affect in Syllogism Performance. *Personality and Social Psychology Bulletin*, 21(8):788–794, August 1995. ISSN 0146-1672. doi: 10.1177/0146167295218001.
- D. R. Morris. Causal Inference in the Social Sciences: Variance Theory, Process Theory, and System Dynamics, 2005.
- M. Okabe and K. Ito. Color Universal Design (CUD) / Colorblind Barrier Free - <http://jfly.iam.u-tokyo.ac.jp/color/index.html>, 2008.
- A. J. Oswald, E. Proto, and D. Sgroi. Happiness and productivity. *The Warwick Economics Research Paper Series TWERPS no 882*, (882):44 pages, 2008.
- B. Parkinson, R. Briner, Reynolds S., and P. Totterdell. *Changing moods: The psychology of mood and mood regulation*. 1996.
- K. Petersen. Measuring and predicting software productivity: A systematic map and review. *Information and Software Technology*, 53(4):317–343, April 2011. ISSN 09505849. doi: 10.1016/j.infsof.2010.12.001.
- B. J. Piper, V. Li, M. A. Eiwaz, Y. V. Kobel, T. S. Benice, A. M. Chu, R. H. J. Olsen, D. Z. Rice, Hilary M Gray, and Shane T Mueller. Executive function on the Psychology Experiment Building Language tests. *Behavior Research Methods*, 2011. ISSN 15543528. doi: 10.3758/s13428-011-0096-6.
- R. Plutchik and H. Kellerman. *Emotion, theory, research, and experience*, volume 3. Academic Press, 1980. ISBN 0125587015.
- J. Russell. A Circumplex Model of Affect. *Journal of Personality and Social Psychology*, 39(6):1161–1178, 1980. ISSN 00223514.
- S. C. D. B. Sampaio, E. A. Barros, G. S. Aquino Junior, M. J. C. Silva, and S. R. D. L. Meira. A Review of Productivity Factors and Strategies on Software Development. *2010 Fifth International Conference on Software Engineering Advances*, pages 196–204, August 2010. doi: 10.1109/ICSEA.2010.37.
- W. Scacchi. Understanding Software Productivity. *Advances in Software Engineering and Knowledge Engineering*, 4:37–70, 1995.

- K. Schwaber and M. Beedle. *Agile Software Development with Scrum*, volume 18 of *Series in Agile software development*. Prentice Hall, 2001. ISBN 0130676349. doi: 10.1109/2.947100.
- T. Shallice. Specific Impairments of Planning. *Phil Trans R Soc B*, 298(1089): 199–209, June 1982. doi: 10.1098/rstb.1982.0082.
- T. Shaw. The emotions of systems developers. In *Proceedings of the 2004 conference on Computer personnel research Careers, culture, and ethics in a networked environment - SIGMIS CPR '04*, page 124, New York, New York, USA, April 2004. ACM Press. ISBN 1581138474. doi: 10.1145/982372.982403.
- D. K. Simonton. Creative productivity: A predictive and explanatory model of career trajectories and landmarks. *Psychological Review*, 104(1):66–89, 1997. ISSN 0033295X. doi: 10.1037//0033-295X.104.1.66.
- P. T. Sowden and L. Dawson. Creative feelings. In *Proceedings of the 8th ACM conference on Creativity and cognition - C&C '11*, page 393, New York, New York, USA, November 2011. ACM Press. ISBN 9781450308205. doi: 10.1145/2069618.2069712.
- I. Stamelos, L. Angelis, P. Dimou, and E. Sakellaris. On the use of Bayesian belief networks for the prediction of software productivity. *Information and Software Technology*, 45(1):51–60, 2003. ISSN 09505849. doi: 10.1016/S0950-5849(02)00163-5.
- S. Wagner and M. Ruhe. A Systematic Review of Productivity Factors in Software Development. In *Proceedings of the 2nd International Workshop on Software Productivity Analysis and Cost Estimation, (SPACE 2008), Technical Report ISCAS-SK LCS-08-08*, number September, 2008.
- D. Watson, L. A. Clark, and A. Tellegen. Development and validation of brief measures of positive and negative affect: the PANAS scales. *Journal of personality and social psychology*, 54(6):1063–70, June 1988. ISSN 0022-3514.
- C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in software engineering: an introduction*, volume 15 of *Software Engineering*. Kluwer Academic Publishers, 2000. ISBN 0792386825.
- J. M. Zelenski, S. A. Murphy, and D. A. Jenkins. The Happy-Productive Worker Thesis Revisited. *Journal of Happiness Studies*, 9(4):521–537, 2008. ISSN 13894978. doi: 10.1007/s10902-008-9087-4.

Appendix

A Experiment Instructions for Participants

Reference Sheet

Hi and thank you for participating to this experiment. This sheet contains the instructions for completing it. First of all, please **do not logout/shutdown/reboot the PC**. We will lose your data otherwise.

Your **Reference Code** is: <Reference Number>

Please, provide it when requested. The experiment is completely anonymous. We only need the Reference Code to connect your surveys with the data that you will provide us during the experiment.

If you have a question, feel free to call one of the supervisors.

The following are the experiment phases.

1. Survey

Please open the browser and go to **<http://goo.gl/gwvvue>** to reach the survey.

Answer to all the provided questions. Remember that the period of time is **the past 4 weeks, including right now**. Provide the Reference Code <Reference Number>/1 (please add /1).

Remember to submit the Survey once you've finished it.

It should take you about 3 minutes to complete it, but take your time.

2. Photographs game

Go to the supervisors and provide your Reference Code <Reference Number>. You will receive two photographs, one at a time. Imagine that you are participating in the Best Caption of the Year contest. This contest is organized by a famous magazine and the winning captions will be published along with the photographs. Your job is to try to win this contest by writing the best captions possible for each of these two photographs. The captions can be absolutely anything you'd like. You can write as many captions as you'd like.

Please, remember to write your Reference Code <Reference Number> in the photographs, too.

This task should take you about 10 minutes, but take your time.

3. Survey

Please open the browser and go to **<http://goo.gl/gwvvue>** to reach the survey.

Answer to all the provided questions. Remember that the period of time is **the past 4 weeks, including right now**. Provide the Reference Code <Reference Number>/2 (please add /2).

Remember to submit the Survey once you've finished it.

It should take you about 3 minutes to complete it, but take your time.

4. Tower of London game

Please open the PEBL software. As **Participant Code** (located in the top-center section of the user interface), enter your Reference Code <Reference Number>.

Do **not** press the "+" button.

On the left side panel, follow this path: *battery/ → tol/ → TOL.pbl*. Select TOL.pbl with the mouse.

Click the button labeled "Add to Chain". The TOL.pbl will appear in the Experiment Chain list.

Click the button labeled "Launch Chain". A new window will appear.

When requested, press key 3 on the keyboard to select **Shallice Test ([1,2,3] pile heights, 3 disks, Shallice's 12 problems)**.

It should take you about 7 minutes to finish the game, but take your time.

When you finish the game, please call one of the supervisors of the experiment. Do not close the program.

We remember you again, please **do not logout/shutdown/reboot the PC**.

Thank you for your collaboration.

B Experiment Instructions for Creativity Judges

The following instructions are provided by (Forgeard, 2011).

Imagine that you are the judge for the Best Caption of the Year contest. This contest has been organized by a famous magazine and the winning captions will be published along with the photographs.

Your job is to rate the captions that have been submitted for each photograph. There is only one criterion for rating these captions: creativity. For each caption, you will provide your mark by expressing your agreement to the sentence “This caption is creative”. You will use a mark that ranges from 1 to 7, where 1 means “I totally disagree” and 7 means “I totally agree”.

I realize that creativity doesn’t exist in a vacuum, and to some extent creativity probably overlaps with other criteria one might apply - aesthetic appeal, organization, richness of imagery, sophistication of expression, novelty of word choice, appropriateness of word choice, humor, for example - but I ask you to rate the captions solely on the basis of your thoughtful-but-subjective opinions of their creativity.

The point is, you are the expert, and you don’t need to defend your choices or articulate a definition of creativity. What creativity means to you can remain a mystery - what I want you to do is use that mysterious expert sense to rate the captions for creativity

C Experiment Instructions for the Problem-Solving Task

The following instructions are provided by (Piper et al., 2011).

You are about to perform a task called the “Tower of London”. Your goal is to move a pile of disks from their original configuration to the configuration shown on the top of the screen. You can only move one disk at a time, and you cannot move a disk onto a pile that has no more room (indicated by the size of the Grey rectangle). To move a disk, click on the pile you want to move a disk off of, and it will move up above the piles. Then, click on another pile, and the disk will move down to that pile.

You will have only a limited number of moves to solve each problem. Before you make your first move, think about the problem to make sure you can solve it within your move limit. If you do not finish the problem within the limit, the turn will end and you will move on to the next problem.

D Scale of Positive and Negative Experience (SPANE)

Mood Assessment Survey

This survey is being conducted by Daniel Graziotin, Pekka Abrahamsson and Xiaofeng Wang of the Free University of Bolzano.

It has been developed so you can tell us how you feel.

The answers you give will be kept private/confidential. No one will know what you write. In addition, your responses written on the survey will be destroyed at the completion of activities related to the analysis of all surveys collected.

Completing the survey is voluntary. Please, answer to all the questions.

Please rest assured that your identity will remain anonymous. The information will not be used to find out your name. No names will ever be reported and you will not be contacted to follow-up on any responses you answered within the survey.

If you did not understand the question, or would like the question repeated, please ask me (your interviewer) to do so.

It should take you about 3 minutes to complete this questionnaire.

***Campo obbligatorio**

I understand the previous Disclosure and I agree with it. I confirm that I voluntarily participate to this survey. *

☐ I agree

Demographics

Reference Code *

This number is provided in the reference sheet. Please write it here.

Nationality *

Gender *

Male

Age *

Input a number (e.g., 23)

Current year of study *

Use 1, 2, 3 for the Bachelor; 4, 5 for the Master.

Scale of Positive and Negative Experience

(C) Copyright by Ed Diener and Robert Biswas-Diener, January 2009.

Please think about what you have been doing and experiencing during the past 4 weeks, including right now.

Then report how much you experienced each of the following feelings, using the scale below.

For each item, select a number from 1 to 5, and cross that number on your response form.

Scale explanation:

1 - Very rarely or never

2 - Rarely

3 - Sometimes

4 - Often

5 - Very often or always

Positive *

1 2 3 4 5

Very rarely or never ☐ ☐ ☐ ☐ ☐ Very often or always

Negative *

1 2 3 4 5

Very rarely or never ☐ ☐ ☐ ☐ ☐ Very often or always

Good *

1 2 3 4 5

Very rarely or never ☐ ☐ ☐ ☐ ☐ Very often or always

Bad *

1 2 3 4 5

Very rarely or never ☐ ☐ ☐ ☐ ☐ Very often or always

Pleasant *

1 2 3 4 5

Very rarely or never ☐ ☐ ☐ ☐ ☐ Very often or always

Unpleasant *

1 2 3 4 5

Very rarely or never ☐ ☐ ☐ ☐ ☐ Very often or always

Happy *

1 2 3 4 5

Very rarely or never ☐ ☐ ☐ ☐ ☐ Very often or always

Sad *

1 2 3 4 5

Very rarely or never ☐ ☐ ☐ ☐ ☐ Very often or always

Afraid *

1 2 3 4 5

Very rarely or never ☐ ☐ ☐ ☐ ☐ Very often or always

Joyful *

1 2 3 4 5

Very rarely or never ☐ ☐ ☐ ☐ ☐ Very often or always

Angry *

1 2 3 4 5

Very rarely or never ☐ ☐ ☐ ☐ ☐ Very often or always

Contented *

1 2 3 4 5

Very rarely or never ☐ ☐ ☐ ☐ ☐ Very often or always

Invia

Powered by [Google Documenti](#)

[Segnala una violazione](#) - [Termini di servizio](#) - [Ulteriori termini](#)

E Experiment Instructions for the Mood Dimensions vs. Productivity Study

Thank you for coming today. I appreciate your participation in this study. I am interested in observing human behaviors during the development of Software. I will now describe you how this study works. I am going to interview you about your demographic data, the project you are working on and the task that you will face today. For about the next 90 minutes, you are going to work on your task. Each 10 minutes, you will be rating your task in terms of how do you feel while working on it. At the end of your task, I will interview you again about how the task went.

There are no right or wrong answers, so simply respond as honestly as you can. Now let me explain your involvement in more detail. You are going to see the following on my tablet <show a paper version of the survey>. You can see 3 sets of 5 figures and a sentence. The three sets of figures are arranged along a scale. You will use these figures to rate how you feel while working on your code. They show three different kinds of feelings: Unhappy vs. Happy, Calm vs. Excited, and Controlled vs. In-control.

The first scale is the unhappy-happy scale, which ranges from a frown to a smile. If you feel completely unhappy, annoyed, unsatisfied, melancholic, despaired, bored, you can indicate this by choosing the figure at the left. The other end of the scale is when you feel completely happy, pleased, satisfied, contented, hopeful. The figures also allow you to describe intermediate feelings of pleasure, by choosing any other pictures.

The calm vs. excited dimension is the second type of feeling displayed here. At one extreme of the scale you feel relaxed, calm, sluggish, dull, sleepy, unaroused. On the other hand, at the other end of the scale, you felt completely stimulated, excited, frenzied, jittery, wide-awake, aroused. You can represent intermediate levels by choosing any of the other figures.

The last scale of feeling that you will rate is the dimension of controlled vs. in-control. At one end of the scale you have feelings characterized as completely controlled, influenced, cared-for, awed, submissive, guided. At the other extreme of this scale, you feel completely controlling, influential, in control, important, dominant, autonomous. If you feel neither in control nor controlled you should chose middle picture.

Your rating should reflect your immediate personal experience, and no more.

The fourth item of the survey is the sentence "My productivity is", followed by an ordered scale of endings for the sentence (very low, below average, average, above average, very high). You should complete the sentence by choosing the appropriate end that describes your current productivity. If you self-assess that your effort on the task is not what you expected, choose "very low". If you self-assess that your effort is a lot more than what you expected, choose "very high". If you think that your effort is what you expected to be, choose "average". You can also choose intermediate values such as "below average" and "above average".

I remember you that there are no right or wrong answers, so simply respond

as honestly as you can. I show you know how the survey looks like on my tablet. You simply touch the picture representing your answer and then press the "Submit" button.

Are there any questions before we begin? Feel free to ask questions during the survey if you have any doubts.

F Mood Dimensions vs. Productivity Study Survey

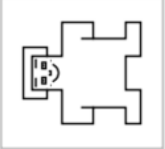
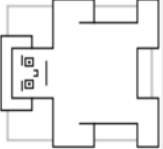
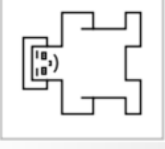
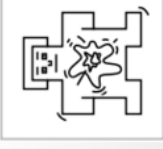
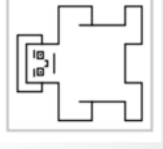
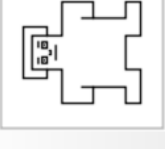
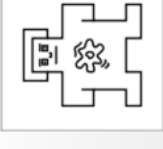
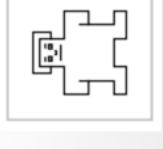
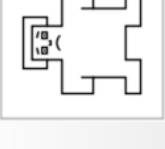
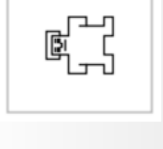
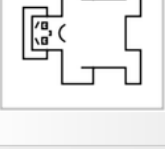
			Very High
			Above Average
			Average
			Below Average
			Very Low
My Productivity is			
Username			Submit

Figure 16: Productivity vs. Mood Dimensions Survey