The Journal of Systems and Software 84 (2011) 739–752

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

A model of job satisfaction for collaborative development processes

Witold Pedrycz [a,1], Barbara Russo [b,*], Giancarlo Succi [b,2]

[a] Department of Electrical and Computer Engineering, University of Alberta, Alberta, Canada
[b] Centre for Applied Software Engineering, Faculty of Computer Science, Free University of Bolzano-Bozen, p.za Domenicani 3, 39100 Bolzano, Italy

ARTICLE INFO

Article history:
Received 22 March 2010
Received in revised form 13 October 2010
Accepted 13 December 2010
Available online 24 December 2010

Keywords:
Job satisfaction
Pair programming
Log linear model

ABSTRACT

Modern software development relies on collaborative work as a means for sharing knowledge, distributing tasks and responsibilities, reducing risk of failures, and increasing the overall quality of the software product. Such objectives are achieved with a continuous share of the programmers' daily working life that inevitably influences the programmers' job satisfaction. One of the major challenges in process management is to determine the causes of this satisfaction. Traditional research models job satisfaction with social aspects of collaborative work like communication, work sustainability, and work environment.

This study reflects on existing models of job satisfaction in collaborative environments, creates one for modern software development processes, and validates it with a retrospective comparative survey run on a sample of 108 respondents. In addition, the work investigates the impact on job satisfaction and its model of the agile practice of Pair Programming that pushes job sharing to the extreme. With this intent, the questionnaire also collected feedback from pair programmers whose responses were used for a comparative analysis. The results demonstrate that Pair Programming has actually a strong positive effect on satisfaction, work sustainability, and communication.

1. Introduction

Collaborative work is the essence of modern software development. In a collaborative environment, developers work in team contributing to common tasks and responsibilities. With collaborative work, knowledge is shared, commonly owned, and rarely lost with team members' turnover. In addition, responsibilities and risks are shared within the team and failures are faster controlled and fixed. In such setting, the quality of the final software product is expected to be high.

The collaborative work requires a strong level of personal commitment and a high degree of involvement in the daily activities, though, and collaborating in team can impact on the programmer's satisfaction and confidence. Competition and subjective norm (Fishbein and Ajzen, 1975) may determine unpleasant pressure causing dissatisfaction at the work. Psychologists define job satisfaction as "present-oriented evaluation of the job involving a comparison of an employee's multiple values and what the employee perceives the job as providing" (Locke, 1984). In team working, perceptions are strongly affected by team members' mutual behaviour. Each day, a programmer needs to mediate his/her with someone else's ideas and behaviour. As such, his/her personal satisfaction depends on the team dynamics and becomes a key indicator of the success of a software project and the achievement of the company's quality goals (Locke, 1984). Therefore understanding reasons of programmers' satisfaction becomes a crucial issue in the management of the software development. This is not a simple issue, though, especially in modern processes of software development in which personal job satisfaction is strictly connected with team performance and team satisfaction.

An example of collaborative work is Pair Programming (PP). PP is a practice of programming – eXtreme Programming (Beck, 2000) in fact – that puts to the extreme the collaborative aspects in a development process (Bryant et al., 2006). For this practice, two developers work side-by-side at one single computer, continuously collaborating on the same design, algorithm, code or test (Canfora et al., 2005; Williams, 2003). PP implies a high degree of job involvement with positive and negative consequences on the management of the software project and the relative developer's job satisfaction (Igbaria et al., 1994).

Traditional literature on social issues in Information Systems (IS) has acknowledged the importance of personal, organization, environmental, and work factors on the study of job satisfaction of IS personnel (Hackman and Oldham, 1980; Goldstein and Rockart, 1984). The intent of our research is to strengthen the above relationships and identify new dependencies between job satisfaction and factors of relationship with co-workers and quality of life in

* Corresponding author. Tel.: +39 0471 016170; fax: +39 0471 016009.
E-mail addresses: pedrycz@ece.ualberta.ca (W. Pedrycz), Barbara.Russo@unibz.it (B. Russo), Giancarlo.Succi@unibz.it (G. Succi).
[1] Tel.: +1 780 492 4661; fax: +1 780 492 1811.
[2] Tel.: +39 0471 016170; fax: +39 0471 016009.

0164-1212/$ – see front matter
doi:10.1016/j.jss.2010.12.018

a collaborative setting. Specifically, it aims at understanding the influence of communication, sustainability of the work and physical work environment on the job satisfaction of developers and it investigates the role of PP on this influence.

Inspired by the work of Goldstein and Rockart (1984), the study introduces a cause–effect model of job satisfaction derived from a comparative analysis carried out on a sample of 108 programmers in international industrial and educational organizations. Data have been collected through an on-line questionnaire. We formulated our research questions in the following way:

R_1: Do communication, work sustainability, or work environment cause job satisfaction of developers?
R_2: Does Pair Programming affect this relation? In particular, does Pair Programming cause developer's job satisfaction?

As a side effect, our research contributes to a challenging open question on the effects of practices of agile methods in industrial settings (Layman et al., 2004). To the best of our knowledge, very few results have been published on this topic (Arisholm et al., 2007).

In Section 2, we introduce the related works that lead to the conceptual model defined in this study. In Section 3, we elaborate on the design of the study and the bias that we addressed. Sections 4 and 5 are dedicated to the questionnaire used and the data analysis and the procedure exploited here. We summarize the main results in Section 6 and conclude the study in Section 7. Section 8 describes future research. Appendix A covers the log linear models and the odds ratio used in the case of two categorical variables. Appendix B is the questionnaire used in this research.

2. Related works

The theory of the Herzberg's two-factors (Herzberg et al., 1959) is considered the ancestor of many of the models of job satisfaction. Herzberg et al. introduce two sets of factors that impact on job satisfaction: the Hygiene factors and the Motivators. The Hygiene factors impact on employees' dissatisfaction whereas the Motivators on the satisfaction. This theory has the merit to have explicitly introduced many of the factors of job satisfaction that are currently used.

Later on, in 1976, inspired by the Herzberg et al.'s work, Hackman and Oldham (1980) have introduced a model of job satisfaction, the Job Characteristics Model (JCM). Although a bit outdated, this model remains the most referenced one for studies on job satisfaction despite later works and research (Soonhee, 2005; Nagy, 2002; Galina and Aleksandr, 2001; Cheney and Scarpello, 1986). The JCM describes the influence of job characteristics on motivation and satisfaction of the single worker under different psychological states. In 1984, Goldstein and Rockart have extended the model including new variables on the relationship with co-workers. Goldstein and Rockart have found that aspects of the collaborative work moderate the cause–effect relation between job satisfaction and job characteristics. For example, the tasks one employee is committed to might determine a different degree of job satisfaction in a collaborative job environment. In addition, the new model of Goldstein and Rockart explains job satisfaction directly from objective aspects of job moving away from the traditional subjective motivations. The model of Herzberg and the one of Goldstein and Rockart will be the references for the present work. In particular, we focus on those aspects that are relevant in collaborative environments. As such, in our work we consider Hygiene factors like Interpersonal Relations (in particular with peers), Working Conditions, and Personal Life and factors of the JCM like Autonomy, Task Significance, and Feedback.

The new methods of development, which emerged in the '90ties gave a strong impulse to the research of job satisfaction in collaborative settings (Nosek, 1998). In particular, agile and open source software development methods have emphasized the value of communication, team working, and quality of the work environment to improve the management of software projects while increasing the job satisfaction of programmers (Raymond, 1999). One of the agile methods also well known in the open source software development, eXtreme Programming (XP), has put this emphasis to the extreme introducing the practice PP (Beck, 2000; Williams et al., 2000; Highsmith, 2003; Arisholm et al., 2007).

In principle, PP enhances the quality of work of the single programmer that can share the burden of the daily goals and responsibilities and increase his/her knowledge and skills through frequent verbal communication (Gittins and Hope, 2001; Nawrocki and Wojciechowski, 2001; Sfetsos et al., 2006; Succi et al., 2001, 2002). As a consequence, job satisfaction increases and turnover of developers reduces. In practice, pros and cons of using PP are still under discussion (Humphrey, 2001). Emery (2002) points out the risk to use atypical practices, like PP, that may turn out to be more expensive than predicted. Using PP may cause adverse attitude and conflicts of personalities that may emerge in a collaborative environment (Sfetsos et al., 2006; Williams et al., 2000). On the opposite, PP decreases the effort for testing and software integration easing the developer's work (Cockburn and Williams, 2001; Canfora et al., 2005).

The effects of using PP have been largely experimented in the educational environment (Katira et al., 2005; Sfetsos et al., 2006; Williams, 2003; Williams et al., 2000; McDowell et al., 2006) that unfortunately is very different from the industry one. Motivation, turn over, and satisfaction of subjects significantly differ in the two environments (Soonhee, 2005; Mannaro et al., 2004). To our knowledge, the paper of (Melnik and Maurer, 2006), is the first studying factors of job satisfaction of agile and non agile teams in a large industrial context. The authors use the Herzberg's theory (Herzberg et al., 1959). In their work, satisfaction results higher in the agile teams because of the ability to influence decisions, the opportunity to work on interesting projects, and the collaboration with users and customers.

2.1. The research model

Typically, a job satisfaction model comes as a cause–effect diagram in which context or research variables moderate the relationship. For example, Fig. 1 illustrates the JCM of Hackman and Oldham.

In the JCM, the first component consists of five core job characteristics that determine job satisfaction: skill variety (i.e., the perceived variety and complexity of skills and talents required to perform the job); task identity (i.e., the extent the job is seen as involving a whole, identifiable task); task significance (i.e., the extent that the job affects the well being of others); autonomy (i.e., the extent the job is seen as allowing for personal initiative in performing the work); and feedback from the job (i.e., the extent that the job, itself, provides information about job performance). The second component consists of psychological reactions to the core job characteristics. The critical psychological states include experienced meaningfulness of work, felt responsibility, and knowledge of results. For example, high autonomy is considered more meaningful by workers as it causes greater feelings of responsibility, and provides clearer understanding of the quality of work. Finally, critical psychological states explain the variability in five specific work outcomes which include job satisfaction. In addition, the motivation of learning moderates the above relation, in the sense that linkages are expected to be significantly stronger for those individuals who are highly motivated to learn and grow on the job.

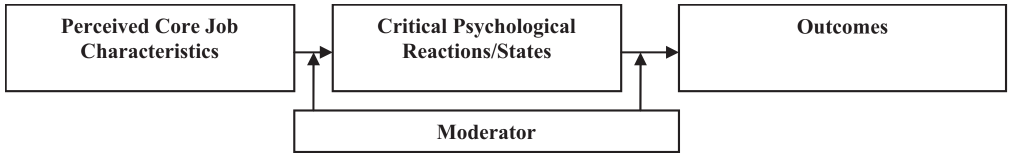


Fig. 1. The major factors in the JCM (Hackman and Oldham, 1980).

Table 1
Professional's job satisfaction factors (Goldstein and Rockart, 1984).

Variable	Definition
Role conflict and role ambiguity	Personal role conflict, inter-sender conflict, and intra-sender conflict
Supervisor leadership and peer leadership measures	Support, goal emphasis, work facilitation, and interaction facilitation
Job characteristics	As in (Hackman and Oldham, 1980)

In the modification of the JCM of Goldstein and Rockart (1984) the outcomes are explained directly based upon aspects of job related to concrete actions and practices of work as in the case of the collaborative work. In particular, in their model, roles and leadership measures are key drivers of satisfaction (Table 1).

Other models consider different variables for job satisfaction, depending on different perspectives and motivations of research (Soonhee, 2005; Cheney and Scarpello, 1986). For example, Scarpello and Campbell (1983) have discussed the significance of vocational and environmental variables in relation to job satisfaction. In particular, they have found that how individuals see their careers along these two dimensions can alternatively explain job satisfaction with respect to the match between needs and rewards. Scarpello and Vandenberg (1992) have further confirmed the clear independence between job satisfaction obtained with fulfilment of vocational needs and the one determined by the job environment with a replication of the original study of Scarpello and Campbell in an industrial environment. Furthermore, not all the variables related to vocational needs, like occupational stability, are relevant in the analysis of job satisfaction (Cheney and Scarpello, 1986).

Starting from the Herzberg's original factors that are relevant in a collaborative environment, keeping the cause–effect with moderation construct of the JCM – the moderator being the use of PP – and adding a perspective related to job environment as in (Scarpello and Campbell, 1983), we define our model of job satisfaction. This model allows us to identify causes of job satisfaction in collaborative environments and discuss the effects of an agile practice on them (e.g. Melnik and Maurer, 2006; Beck, 2000; Williams et al., 2000).

To select the factors for our causal model we have considered three major sources of information:

(1) The literature on collaborative work, (Keller, 1983; Martin and Hunt, 1980; Müller and Price, 1990; Nicholson, 1977; Price and Müller, 1986; Brooke and Price, 1989; Hackman and Lawler, 1971). Brooke, Hunt, Keller, Martin, Müller, and Price have found a positive influence of group cohesion on job satisfaction and negative influence on turnover or absenteeism. Hackman and Lawer have investigated factors of job satisfaction as "dealing with others" the degree to which job requires employee to deal with other people or "friendship opportunities" the degree to which a job allows employees to talk to each other on the job and establish informal relationships at work. The majority of the factors of their model of job satisfaction are the ones used in the model of Hackman and Oldham (1980).
(2) The studies on job satisfaction in an industrial context including the ones we have mentioned above and other like (Galina and Aleksandr, 2001) or (Scarpello and Vandenberg, 1992). For example, Galina and Aleksandr found that employees that fear to loose their jobs have higher satisfaction rates and attitude to work. Again in terms of employees' perceptions, Scarpello and Vandenberg discuss the discrepancy between what employees expect from their job and what actually they get in relation of job satisfaction. This paper introduces one key point: employees' perception can modify the actual cause–effect relation of a model of job satisfaction. We discuss this in the section on demographic variables.
(3) The research papers on the agile development (e.g. Beck, 2000; Cockburn, 2006). Unfortunately, there is not much empirical literature on the effects of PP on job satisfaction (Dybå and Dingsøyr, 2008). As such, our work also aims at contributing to this. There are few observational studies that we will discuss while introducing causes and sub-causes of job satisfaction, but to our knowledge the only two papers that discuss the Herzberg's or the JCM models in the agile collaborative context is the one of Tessem and Maurer (2007) and the one of Melnik and Maurer (2006). The first reports of an assessment analysis of JCM in the case of Scrum. Scrum is another example of agile method (Schwaber, 2004). The analysis is based on unstructured interviews and experience reports and it does not assess the cause–effect relation of JCM. The second discusses the Herzberg's factors in a large industrial context in terms of the roles and the experience in the agile practices. The work is abased on a large survey, but does not discuss any causal relation with job satisfaction.

From this investigation, we come up with three macro causes of job satisfaction in a collaborative setting:

2.1.1. Communication

Knowledge transfer is a key issue in any development methodology. Factors here concern communication as human interactions. They are inspired by the Hygiene factor "Interpersonal Relations" (specifically, "Interpersonal Relations with Peers," Herzberg et al., 1959) and "Feedback" of JCM (Hackman and Oldham, 1980) and considered in a collaborative environment. Herzberg refers to Interpersonal Relations as factor of dissatisfaction, in the negative meaning. In our study, we consider communication more as a Herzberg's Motivator related to Interpersonal Relations and synonymous of feedback as in JCM. In particular, we have considered communication among developers and among units to capture the micro and the macro communication within an organization. Macro communication is also relevant to reveal "Feedback"(Hackman and Oldham, 1980). A second aspect we have considered is how communication is organized. In collaborative software development, meetings are essential for the progress of the activities. In particular, XP meetings are scheduled on a daily base with stand-up meetings that help to spare developers' time (Beck, 2000) and through continuous interactions as in PP. PP emphasizes sharing knowledge and responsibility of work through communication (Beck, 2000; Williams et al., 2000; Janes et al., 2003; Cockburn, 2006). A few empirical works report of the explicit effect of PP on job satisfaction, communication and their mutual relation (Cockburn and Williams, 2001; Domino et al., 2003; Melnik and Maurer, 2002, 2006). Cockburn and Williams have found that PP

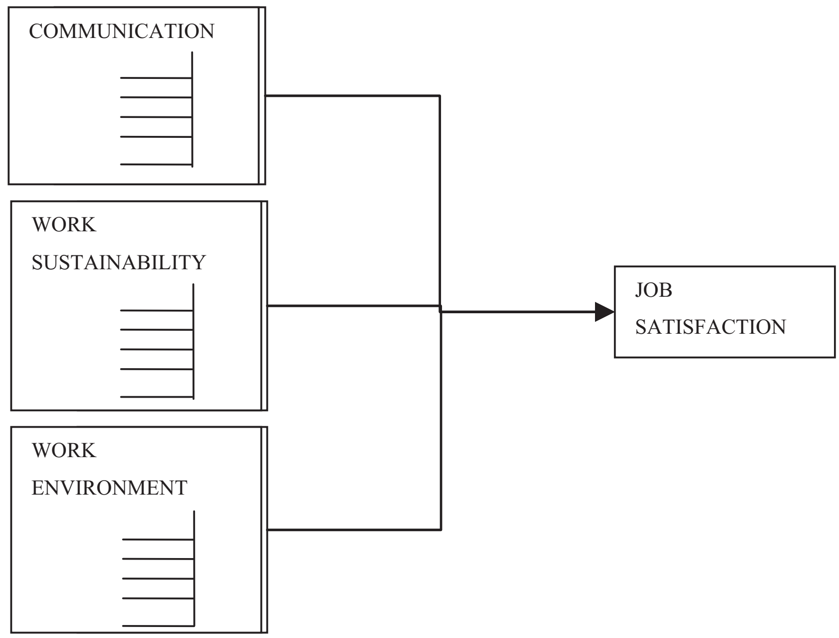


Fig. 2. The conceptual model derived from the literature. Various items compound each factor.

enhances team communications and satisfaction. Domino et al. (2003) have analyzed why PP increases developers' satisfaction. In teams that adopt PP, there is a positive correlation between type and amount of interactions between the developers – that they called faithfulness to the method – and job satisfaction, in that higher and better communication implies higher job satisfaction. In a comparative study of Melnik and Maurer (2006), the surveyed students reported that working with PP helped them to develop professional skills such as communication. In 2002, Menelik and Maurer also reported that test-first design was difficult for many interviewees. The authors believe that this is because design in itself is very difficult, and writing tests first forces students to make design decisions early when they have little experience of the system. As such, in the present study we also include "communication of design changes" as a key specification of communication in collaborative software development processes.

2.1.2. Work environment

The physical environment may affect the satisfaction of the developers. This cause is related to the Hygiene factor "Working conditions," (Herzberg et al., 1959). For example, office layout, arrangements of the furniture, noise, and roommates can influence the motivation to work of programmers (Gallis et al., 2003). In PP, work conditions can be even be more crucial as the pairs need to share tables, screens, and rooms. (Beck, 2000; Williams et al., 2000). Beck claims "If you do not have a reasonable place to work, your project won't be successful." Mannaro et al. (2004) claim that employees that use XP have greater job satisfaction feeling that the job environment is more comfortable. Gallis et al. (2003) illustrate as empirical studies that investigate the effects of relocating employees from traditional offices to open offices – typical of XP and of team collocation – show reduced satisfaction with the open physical environment, increased physical stress, deterioration of team member relations, and lower job performance among the employees (Brennan et al., 2002). A similar study of exposure to simulated low intensity open-office noise concluded that the noise did not result in elevated stress, but lowered the task performance, which is indicative of reduced motivation (Evans and Johnson, 2000). Office layout for PP is also discussed by Müller and Tichy (2001) whose results suggest relaxing the rule of single screen sharing.

2.1.3. Work sustainability

Factors here concern how individuals manage interactions and responsibilities between home and office and the daily workload. These factors are related to the Hygiene factor "Personal life" (Herzberg et al., 1959) and "Autonomy" and "Task Significance" of JCM (Hackman and Oldham, 1980). Such factors are widely acknowledged in research works on stress, job involvement, and satisfaction of IS professionals and software developers (Rajeswari and Anantharaman, 2003; Stoeva et al., 2002; Sonnentag et al., 1994; Hackman and Oldham, 1980). In addition, sustainability of the work is one of the major principles of XP, as an agile method (Highsmith, 2003). In particular, Sonnentag et al. (1994) have focused on factors like workload (factor 9) and balance between work and home (factor 4) in the analysis of burnout of software developers. In our model, we also considered monotonous tasks has they determine low "Task significance" (Hackman and Oldham, 1980). Sharing responsibilities and ownership of the final work can help the sustainability of the work and the satisfaction of the developers adopting XP (Higman et al., 2003). Specifically, this is a positive outcome expected with the use of PP (Acuña et al., 2009). On the other side, personality conflicts and the rule "40 hours a week" increase anxiety and stress in team adopting PP also decreasing job satisfaction (Domino et al., 2003; Dybå and Dingsøyr, 2008; Acuña et al., 2009).

Taking all these aspects into account, we summarized the final conceptual model in Fig. 2.

In each major category, we identify a set of sub-factors, as illustrated in Table 2. Each sub-factor is a variable of the model and, in principle, influences developer's job satisfaction differently. Variables might be mutually dependent resulting in an undesired redundancy in the model, though. For example, work environment and communication might depend on each other as good communication is facilitated by comfortable work conditions. If this is the case, developer's job satisfaction is effected more by "communication in a good environment" than by "communication" and "good

Table 2
Causes of job satisfaction and their sub-factors.

Communication (CM)	Work environment (WE)	Work sustainability (W)
Communication between departments is satisfactory (CBDp)	Workspace and office layout is good (WL)	Is not too monotonous (TM)
Communication between developers is good (CBD)	Lighting is good (L)	Sustainable amount of work (SAW)
Design changes communicated quickly (DC)	Noise is good (N)	Manage to balance between home and work (BHW)
Meetings well organized (*M*)	Heating is good (H)	Is not source of worry and stress (SWS)

environment.” In Section 4, we perform an inferential analysis and log linear regression to remove the exceeding factors.

3. The design of the study

To collect our data we designed and published an on-line questionnaire. As suggested in (Schuman and Presser, 1981; Judd et al., 1991) we used a pre-survey on a small sample of developers (published in Succi et al., 2002) to base the design our final questionnaire (Appendix A).

Following the theory of experimental design (Campbell and Stanley, 1963), we developed a *simple survey design*, which belongs to the techniques of *retrospective analysis*. The simple survey design offers to the respondent the opportunity to classify him/her as having or not been exposed to a given treatment. For example, in our case the respondent is classified as “using pair programming” or “not using pair programming” as the treatment corresponds to the use of Pair Programming.

As we collected the responses we have divided them in two groups depending on whether the respondent used Pair Programming. At the end of the data collection we have formed a big sample – that we use for the validation of the model in Fig. 2 – and two equal size groups in which the sample is divided – that we use for the comparative analysis on the use of Pair Programming. Following the notation of the experimental design, in the comparative analysis we have called *treatment group*, the group that uses Pair Programming, and *control group*, the other.

We need to notice here that we have had no control on the variables of our study as we have observed them only ex-post. Hence the cause–effect relations studied in this paper may have been exposed to an undesired bias jeopardizing the internal validity of the analysis (Campbell and Stanley, 1963; Frigon and Mathews, 1997). Aware of this, we have decided to use techniques of quasi-experimental design to control the bias. These techniques aim at balancing the lack of control on the collection of the data with a fine grained design of the data format and the instruments to collect them and an accurate choice of the statistical analysis performed afterwards. In what follows, we describe this approach.

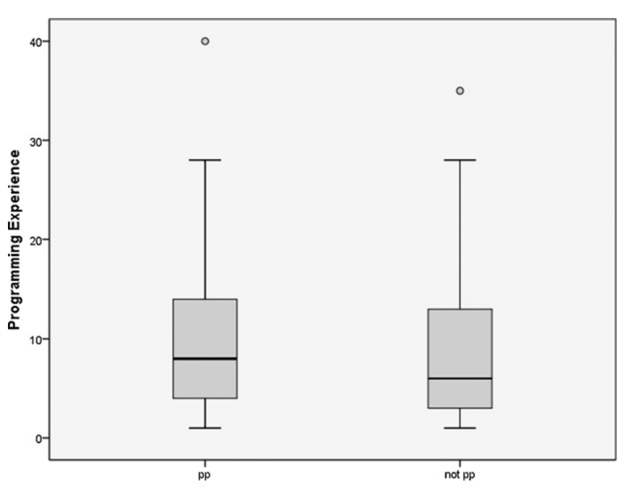


Fig. 3. Programming experience in the two groups.

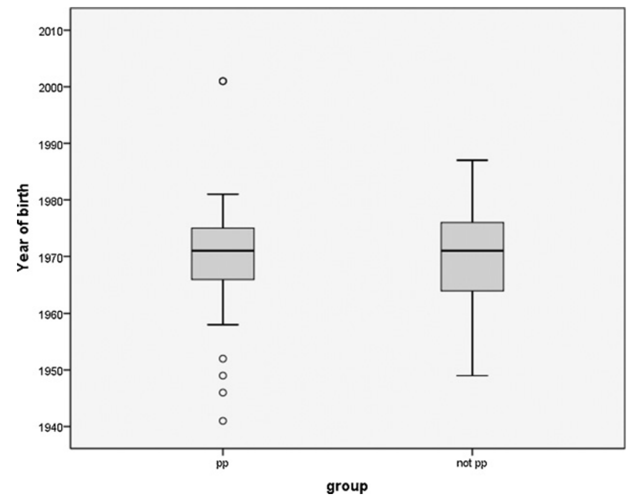


Fig. 4. Age in the two groups.

We have identified the following threats to the internal and external validity of the analysis:

a. Selection. No randomization process for the membership to a group of the sample was possible as respondents selected the group on their experience
b. Representativeness of the sample. No full control on the process of information that has led to volunteers' inscription: the questionnaire might not have reached all the representatives
c. Maturity and selection of the respondents. Different levels of job satisfaction in the two groups that may be underneath the motivation to participate to the survey
d. Misconception of pair programming
e. Misunderstanding of some questions

To control each of these factors of bias we have used the following quasi experimental techniques:

a. We have structured the questionnaire with extra information to control and rule out plausible factors rival to the analysis. This follows the idea of the Retrospective Pre-test (Campbell and Stanley, 1963): no difference in threats to internal validity may increase the plausibility that there is no difference in the two groups of the sample. Thus, in the questionnaire we have inserted a section on demography and curriculum and we have run homogeneity tests on the two groups (Section 5.2).
b. In the analysis we have used a statistical randomization on data. Namely, we have re-sampled our set of data to consolidate results and assure the internal validity of our analysis.
c. To validate the proposed model we have used the holdout method (Kohavi, 1995).
   a. During the collection of the responses, we have publicized the questionnaire worldwide and through all the standard channels of software engineering.
   b. We have selected the sample on the base of the response arrivals and we have tested it for demographic and curriculum heterogeneity.

c. We have controlled the representativeness of our sample retrospectively with the homogeneity tests with respect of the population of the ACM members.
d. This is related with the lack of a randomization process and the actions taken are the ones in 1. In the questionnaire for pair programmers, we have added demographic and subjective questions to control the degree of knowledge of the practice.
e. Misunderstanding of some questions is typical of an on-line questionnaire when information is provided only virtually. To control this bias we have selected questions on the base of a pre-survey and previous existing literature (Succi et al., 2002; Schuman and Presser, 1981; Judd et al., 1991). Where possible, we have structured the questionnaire in multiple choice short questions.

4. The questionnaire

As we mentioned, our data has been collected through an on line questionnaire to reach the largest possible heterogeneous group of developers. Respondents have been recruited through the standard channels of the Software Engineering community: conference announcements (like ICSE, OOPSLA, XP conferences) mailing lists (like SEWORLD and SEA), newsgroups (like comp.software-eng) or private network of software engineers.

Two PHP pages have been published on-line, each containing a questionnaire. Both questionnaires have a common core part concerning demographic aspects and aspects of collaborative work and a section on the use of design review, code review and unit testing. The PHP page of the pair programmers has an extra section on the knowledge of PP.

The questionnaire has been designed in order to obtain information on the three macro categories identified in the literature. From the pre-survey, performed on a smaller set of programmers (Succi et al., 2002), we have derived the factors and the composition illustrated in Table 2.

As in traditional literature of job satisfaction (Hackman and Oldham, 1980; Goldstein and Rockart, 1984), the questionnaire also includes a section on respondents' attitude at work, which is not entered in our model as it correlates of job satisfaction. They are used, instead, to compare the two groups in terms of their expectations. Thus, we treat them as demographic variables, to control the homogeneity of the two groups. The questionnaire is in Appendix B.

5. Data analysis

The sample consists of two groups of 54 respondents, selected by order of response arrivals: the group using Pair Programming (PP group) and the one using another method (NPP group). The 108 respondents belong to 83 different organizations including 8 universities, 28 small software houses, 16 medium size software houses, 15 large software departments or software houses, and 16 other private companies.

In the following, we discuss the representativeness of the whole sample, the homogeneity in the two groups, and the level of knowledge of Pair Programming.

5.1. Representativeness of the sample

In order to determine the representativeness of our sample, we compare the demographic variables with the ones reported in the ACM membership profile report of the same year in which we collected the answers, 2006.[3] In our sample, we find: a similar percentage of men – 87% of our sample compared to 86% in the ACM report – a lower percentage of academia staff – the 8% of our sample against the 20% in the ACM report – a similar graduate experience – 84% of our sample has a bachelor degree against the 79% in the ACM report. Different is the geographical location of the sample. In our case, the sample is more distributed across the countries. In the case of age, our sample is a bit younger than the ACM one as we have about 3% less than 25 years old 90% between 25 and 50 years old and about 8% above 50, but with a good level of experience with about 4% with less than 2 years of experience, 35% between 5 and 10 years of experience, 40% with more than 10 years of experience.

This data provides enough evidence that the demographic characteristics of our sample are similar to those of the total ACM population. As ACM members represent a wide heterogeneous population of professionals in Computer Science, we believe that the similarities found give some support to the representativeness of our sample.

5.2. Homogeneity of the two groups of the sample

To control for homogeneity in the two groups of the sample we have analyzed the demographic variables (Section 5.2.1), the maturity of the respondents in the two groups (Section 5.2.2), and the perceived causes of satisfaction (Section 5.5.2).

5.2.1. Demographic variables

To control possible bias, we have tested the two groups for homogeneity in the variables Gender, Age, Programming Experience, and Nationality. For Gender, a simple descriptive analysis is sufficient. The PPG consists of 47 males and 7 females whereas the NPP group has 48 males and 6 females. The gender is therefore similarly distributed in the two samples. For Nationality we have 22 countries. The countries more represented are United Kingdom, Italy, Germany, and United States of America. For these countries, the difference in the two groups is minimal. The other countries have 1–5 respondents. For Age and Programming Experience the box plots of the distributions are similar in the two groups (Figs. 3 and 4). In addiiton, the index of dissimilarity, (Agresti, 2002), of the two sample groups is below 0.061 for each variable. Both results indicate the homogeneity of the variables in the two groups and a level of dependability of our analysis.

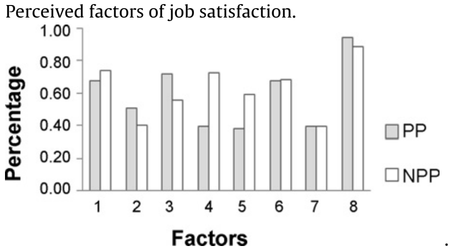
5.2.2. Attitude toward socialization in the two groups

One of the major statements, about Pair Programming is that pair programmers are in principle more enthusiastic and motivated to work in team as they have a specific attitude toward communication and they embrace the practice as a philosophy more than a paid work (Dybå et al., 2007). As such, they might be more satisfied in their job. For such reasons, we have asked the respondents about their perceived factors of job satisfaction.

Table 3 shows that the answers of the two groups are similar. In particular, "Working in a good team" is the most relevant perceived cause of job satisfaction in both groups. This implies no perceived difference in socialization qualities between the two groups. No difference in "Salary" and "Work Environment" too. There is a small difference in communication that favours of the Pair Programming group. Such difference pertains to the communication outside the work team, though. The major difference is that the Non Pair Programming group focuses more on tasks, projects and see the work done.

[3] http://www.campus.acm.org/public/pressroom/about_acm/membership_profile.cfm?CFID=692674&CFTOKEN=34803144.

Table 3
Perceived factors of job satisfaction.



#	Perceived factors of job satisfaction
1	See finished product
2	Communication with management
3	Communication with other employees
4	Task
5	Project
6	Work environment
7	Salary
8	Working in a good team

With these results we can easily assume that the enthusiasm of Pair Programmers does not significantly impact on our research.

5.3. Knowledge of pair programming

As a misconception of Pair Programming may bias conclusions, we have administrated two extra sets of questions: to the PP group only, a set of questions on the major facts about PP and to the whole sample, a set of questions on the use of design review, code review, and unit testing as these are development's common instruments that are commonly adopted in PP (Beck, 2000).

In Table 4, we report ten facts about PP and the answers with highest percentage of respondents. These percentages confirm that the PP group has a good knowledge and confidence in the practice.

The analysis of the use of design review, code review, and unit testing reports that the NPP group does not use any of the three practices while all of them are in use in the PP group. Unit Testing is the most used practice in the PP group – 95% of the respondents. This difference further validates the membership of the respondents: unlike the PP group, the NPP group consists of developers that do not have practice on the most known instruments used in Pair Programming.

Table 4
Level of knowledge and use of Pair Programming in the PP group.

Facts about PP	Choice with the highest percentage	Percentage
How long have you been PP?	More than one year	54
What is the role of the person not typing	Perform continuous design/code review	90
I am confident in working with PP	Strongly agree/agree	85
When I pair programming I enjoy my job more than when I solo programming	Strongly agree/agree	86
How much time should partners work separately?	Less than 30%	51
What is your opinion on PP?	PP is more effective than non PP	69
How important is continuous customer's feedback?	Important	49
Is the quality of the code higher when you pair programming	Strongly agree/agree	98
Is the stand up meeting important?	Strongly agree/agree	89
Does PP enhance communication in the team?	Strongly agree/agree	100

5.4. The proposed statistical method

We use a three steps analysis to refine and validate the model shown in Fig. 2: (1) dependency analysis, (2) dependency analysis with moderation – being the moderator "use of PP" –, and (3) log linear analysis of the correlates of job satisfaction in the two sample groups.

Step1. The dependency analysis investigates the relation among the variables of our theoretical models. The aim of this analysis is to reduce redundant variables and to determine the relations of the remaining ones with job satisfaction. For this purpose, we perform the following:

a. To reduce redundancy: a 1-tail Spearman's cross correlation analysis for multiple testing among all the sub factors of Table 2.
b. To determine dependency with job satisfaction and validate the model: a 1-tail Spearman's rank correlation of the sub factors in Table 2 with job satisfaction.

Step2. The dependency analysis with moderation aims at quantifying the influence of a moderator variable on the dependency found at Step 1. In our work, the moderator variable identifies the membership of a subject to a sample group. The results of this analysis tell us whether the membership to a group affects the patterns found with the dependency analysis of Step 1. In this case

a. We define a dummy variable, Group, that identifies the membership to a group of the sample
b. Then we run a partial correlation analysis with the moderation of the variable Group

Step3. The log linear analysis of the correlates of job satisfaction defines a model describing the degree of dependency among all the significant variables identified in Step 1 and Step 2 – including job satisfaction and Group. The model is used to compare the probability of occurrence of given combinations of values of the variables in the two sample groups.

a. With a stepwise procedure, we identify the best fit generalized log-linear model of the contingency table of Group and independent sub-factors determined in Step1 and Step2
b. Then we validate the model in (a) with the holdout method (Kohavi, 1995) to assess the robustness of the model
c. Finally, we compute the odds ratios with respect to Group of the log-linear model determined in (a) to compare the probability to have certain answers among the model variables in the two groups

Data has been analyzed with the SPSS statistical tool. If not stated otherwise, all the significance levels have been set as 0.05. When needed, we will use the False Discovery Rate (FDR) to control for multiple testing. This method corrects the significance of each test by a factor $(m+1)/2m$ where m is the number of (in)dependent tests (Benjamini and Hochberg, 1995; Benjamini and Yekutieli, 2001).

5.4.1. The log linear models of contingency tables

The variables of this study are categorical, have ordinal scale and multinomial distribution that suggests the use of the non-parametric statistics. Thus, we use the Spearman non-parametric correlation and the log-linear regression modelling analysis for contingency tables – a standard approach for categorical data (like two threads of a die), retrospective studies (Agresti, 2002) and data coming from observational experiment (Lloyd, 1999) where no randomization process in the sample selection is possible. Log linear models have been widely used in social, political and natural sciences (Lakhan and Lavalle, 2002; Agresti, 2002).

Log linear models capture various levels of correlations of variables. Therefore these models can tell which are the variables that are correlated with job satisfaction and whether there are other variables interfering in this correlation. The log-linear models predict the expected values of cells in contingency tables of categorical variables. In Appendix A, we illustrate the case of two categorical variables.

A log linear model is outlined by a design that represents the dependency among the variables. There are two types of models, saturated or customized. The difference consists in the design of the model. The saturated models (M_0) are complete, they contain all the possible terms of multiple interactions among the model's variables. A saturated model M_0 in four variables has the design

$$(X*Y, X*Z, X*W, Y*Z, Y*W, Z*W, X*Y*Z, W*Y*Z, \ldots, X*Y*Z*W)$$

where the * operator represents the correlation among variables. A simple correlation X^*Y defines the dependency of two variables; a higher order correlation (e.g. X^*Y^*Z) identifies the impact of other variables (e.g. Z) on a simple correlation of two variables (e.g. X^*Y).

The design is a way to represent the stochastic equation underneath the logarithmic value of the expected mean of the Poisson or Multinomial probability distribution of a cell in a contingency table (Agresti, 2002). The design represents only the type of correlations in the model not the intensity of these correlations. The customized model (M) is not complete and in its design some correlations are zero (e.g. (X^*Y, X^*Z, Z^*W, X^*Y^*Z)).

To fit our data we use regression techniques on the log-linear models. To select the best fit log-linear model we use the typical twofold procedure:

Selecting the design of the model. For different choices of non saturated models M we test the null hypothesis

$$H_0 : M = M_0$$

with both Likelihood Ratio (LR) or Pearson. For low p-values (for LR $p < 0.10$ and for $p < 0.05$) the null hypothesis is rejected and the two models are considered different. As the saturated models contains all the dependency relations among our variables, rejecting the Null Hypothesis means that we cannot substitute the saturated model with a simpler model, thus all the correlations not included in the customized model M are in theory non zero and need to be considered in the model. Therefore the saturated model M_0 is selected. If the p-value is higher, the Null Hypothesis cannot be rejected. This means that the correlations excluded by the customized model M can be zero. In this case, we choose the customized M as its design is simpler and shows less dependency (Agresti, 2002; Lloyd, 1999).

$$\text{Odds Ratio of } Z = \log\left(\frac{Pr(Z=2; X=i; Y=j)}{Pr(Z=1; X=i; Y=j)}\right)$$

Computing the parameters of the model. We use the Maximum Likelihood Estimation (MLE) to estimate the model parameters that expressed the intensity of the correlations in the selected model (Lloyd, 1999). Once they have been determined, we compute the odds ratios of the log linear model (Agresti, 2002). Odds ratios are used to evaluate the difference in conditional probability of the dichotomous variable in its two values. For example, if Z is a dichotomous variable with values 1 and 2, the odds ratio of Z with respect to two variables, X and Y, for values i and j is:

Odds ratios less than 1 indicate response values with higher probability in the group with $Z = 1$. Odds ratios greater than 1 mean response values with higher probability in the group with $Z = 2$. Odds ratios equal to 1 mean no difference of probability in the two groups.

What we consider in our study, are the conditional probabilities of the dichotomous variable defined by the membership to a group. This is a standard approach often used in polls to compare probabilities of events in two groups. In our case it will compare the sub factors' values in the two groups.

5.4.2. Definition of the model variables from the questionnaire

From the questionnaire we extract 30 variables: 4 supporting the identification of the sample, 8 on expected factors of satisfaction, 10 restricted to the respondents using PP, 3 on project review (code, design and unit testing), and 13 on job satisfaction, work expectations, work sustainability, communication, and work environment.

The variable for job satisfaction (JS) is defined by a direct question "How do you feel about your job?" while those for work, communication and work environment have a set of detailed questions whose average defines the corresponding variable (Table 2).

As we mentioned, the membership to a group of the sample is described by the dichotomous variable Group. The variable Group assumes value 1 for the Pair Programming Group (PP group) and 2 for the Non Pair Programming Group (NPP group).

The variables Work sustainability (W), Work environment (WE), Communication (CM) are defined as the average of their sub factors, whose values vary in a Likert scale from "Strongly Agree" (5) to "Strongly Disagree" (1) Several studies have shown that global variables are prone to bias (Allen and Yen, 2002; Scarpello and Vandenberg, 1992) and sub-variables are more reliable. To understand the different information provided by global and sub-variables, we have computed the Cronbach's alpha reliability score of the factors of each global variable. This score measures how well a set of variables represents a single unidimensional latent construct (a global variable). Cronbach's alpha can be written as a function of the number of variables the average inter-correlation among them. In our study, this has not resulted in a percentage over 70% being the typical threshold. Thus, we have decided to consider both global and sub-factor variables.

5.5. Investigating the dependencies with job satisfaction and the moderation of the "use of PP"

In this section, we perform a dependency analysis in two variables. As all our variables are categorical and have non-parametric distribution, we use a bivariate 1-tail Spearman's rank correlation between each variable and Group.

5.5.1. Job satisfaction in the two groups

The histogram shown in Fig. 5 indicates that both groups have preferably answered toward satisfaction and the PP group shows twice as many responses as the NPP group in the case of high satisfaction.

5.5.2. Factors of job satisfaction—dependency analysis

In this section, we discuss the factors influencing JS and their interrelations.

As the first step, we run a multiple cross correlations with JS using the Spearman rank test. Hence, we set $\alpha = 0.05$ and correct this significance level for single test with the FDR, α correction = 0.031 (Benjamini and Hochberg, 1995). We have chosen the FDR for dependent multiple tests (Benjamini and Yekutieli, 2001) as we have noticed some dependencies in the first run of the correlation test.

Table 5 shows that a model of job satisfaction is in fact defined on work sustainability and communication (Fig. 6) as work environment (WE) does not correlate with JS.

As the Cronbach's alpha test does not guarantee internal consistency of the major variables, we also need to analyse the way in which each sub-variable contributes to JS. To do it, we perform a correlation analysis with JS (Table 7, second and eight columns).

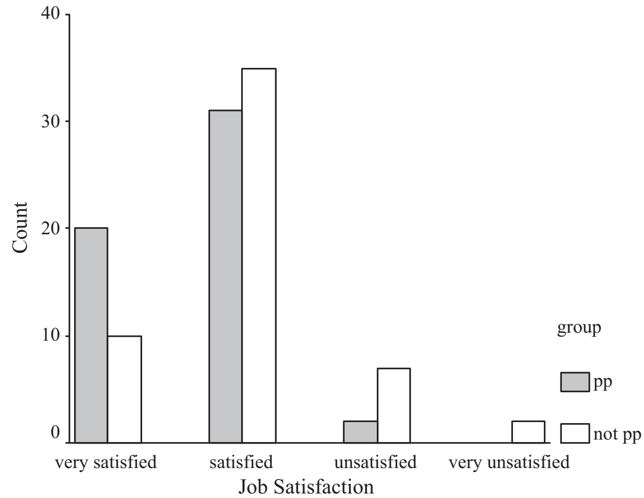


Fig. 5. Job satisfaction in the two groups.

Table 5
Significant correlations among the major variables and Group, $\alpha = 0.031$.

Spearman correlations $p < 0.031$	Group	JS	W	WE	CM
JS	0.268	1	0.65		0.35

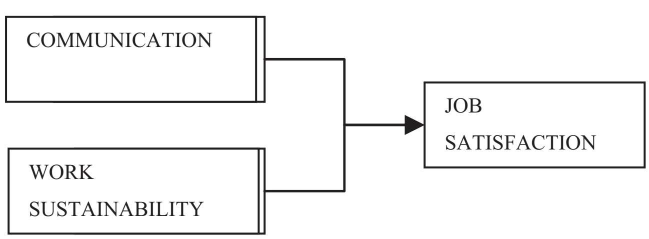


Fig. 6. Model of job satisfaction on the whole sample.

Table 6
Significant cross correlations among sub-variables of W and CM (α correction = 0.027).

Corr.	W		
	SAW	BHW	TM
CM			
CBDp	0.298 $p = 0.001$	0.254 $p = 0.008$	
CBD			0.231 $p = 0.017$

Table 7
Significant Correlations among sub-variables and JS (α correction = 0.027).

(a)

Corr.	JS	W			
		TM	SAW	BHW	SWS
JS		0.640 p = 0	0.263 $p = 0.003$		0.306 $p = 0.001$
TM	0.640 $p = 0.000$		0.298 $p = 0.001$		0.270 $p = 0.002$
SAW	0.263 $p = 0.003$	0.298 $p = 0.001$		0.327 $p = 0.000$	0.324 $p = 0.000$
BHW			0.327 $p = 0.000$		
SWS	0.306 $p = 0.001$	0.270 $p = 0.002$	0.324 $p = 0$		

(b)

Corr.	JS	CM			
		CBDp	CBD	DC	*M*
JS		0.215 $p = 0.0011$	0.338 $p = 0.000$		0.281 $p = 0.003$
CBDp	0.248 $p = 0.001$				0.292 $p = 0.002$
CBD	0.337 $p = 0.000$			0.398 $p = 0.000$	0.438 $p = 0.000$
DC			0.398 $p = 0.000$		0.344 $p = 0.000$
M	0.281 $p = 0.003$	0.292 $p = 0.002$	0.438 $p = 0.000$	0.344 $p = 0.000$	

Table 8
Significant correlations among 12 factors with Group, α correction = 0.027.

Corr.	W	CM		
	SAW	CBD	DC	*M*
Group	0.337 $p = 0.000$	0.301 $p = 0.002$	0.308 $p = 0.001$	0.341 $p = 0.000$

Table 9
Significant partial correlations with JS w.r.t. Group, α correction = 0.027.

Part. corr.	W			CM
	TM	SAW	SWS	CBD
JS	0.582 $p = 0.000$	0.369 $p = 0.000$	0.352 $p = 0.000$	0.338 $p = 0.000$

We also perform a cross correlation between the 12 sub-variables to measure their interdependencies. Significant correlations are displayed in Tables 6 and 7. No correlation of WE and its sub-factors appears.

The correlations with JS in Table 7 determine a further refinement of our initial model on the whole sample (Fig. 7).

In particular, two sub-factors of communication and three of work sustainability are significantly correlated with JS. We exclude "Meetings well organized" from the model as it correlates with all the sub-factors of communication (Table 7).

If we wanted to use the results of correlations with JS to further exploit it in a linear regression analysis we would have encountered a significant problem: sub-factors of communication and work sustainability in Fig. 7 are highly cross correlated (Table 6). This indicates probable collinearity among the various candidate regressors of an eventual linear model of JS. As collinearity is not advisable and removing it would have been resulted into a poor linear model, we have decided to use log linear models that more suitably exploit the dependency with JS in a non-linear fashion.

5.6. Investigating the variables' variation with Group

The analysis presented here concerns the influence of the variable Group on the model of Fig. 7. We use correlation analysis and log linear models to investigate this issue as in the following.

5.6.1. Correlation analysis with moderation

We correlate the variables with Group to determine whether the model in Fig. 7 is sensitive to the use of Pair Programming. Again, Work Environment and all its sub-factors result not relevant (Tables 5 and 8).

The partial correlations in Table 9 confirm the zero correlations with JS of Table 7 with the exception of CBDp and *M*. According to these results, the correlation between JS and CBDp or *M* is no more significant if we eliminate the contribution of Group. This result

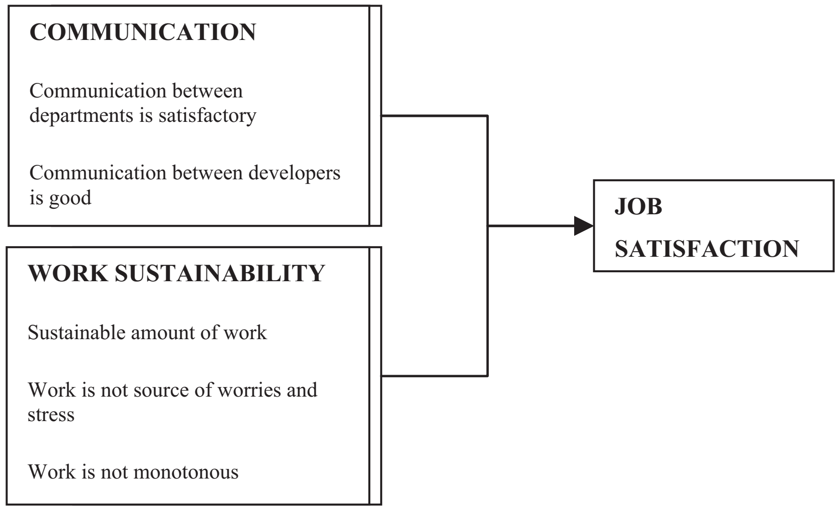


Fig. 7. The conceptual model refined and validated on the whole sample of respondents.

confirms that the cause–effect model of Fig. 7 is not influenced by the membership to a group with the only exception of “Communication between departments is satisfactory” and “Meeting well organized” which means that the relation with job satisfaction is significant only because either job satisfaction or the factor is correlated with Group (Tables 5 and 8).

5.6.2. Log linear analysis

Here we perform a multivariate analysis using the log linear models to understand the total degree of the dependence found in Step 1 and 2.

To find the design of the log linear model, we use the “entry and backwards elimination” multivariate method with probability for removal at 0.05 (Agresti, 2002). After running the statistic algorithm, the design turns out to be

$$(\text{Group} \times \text{DC},\ \text{Group} \times \text{SAW},\ \text{Group} \times M,\ \text{Group} \times \text{JS},\ \text{DC} \times M)$$

with goodness of fit of the Likelihood ratio and the Pearson tests equal to 0.491 and 0.760, respectively (both significant values at 0.05; Agresti, 2002, p. 186).

To further validate the model and to add randomization to our data we use the holdout method (Kohavi, 1995) splitting the sample 60–40%. The method is used to control for over-fitting, had we considered too many variables in our original model. The method determines a best fit model on the 60% training group of the data and validates it on the remaining 40%. This is repeated ten times. The splitting 60–40 is quasi-random: among the random splits, we select the ones with homogeneous distribution of the variable Group. We use an “entry and backwards elimination” method on the training group to identify a model; then we perform a goodness of fit test on the testing group to compare the model with the saturated one. We repeat the procedure ten times and we rank the ten outcome models in term of the significance of the goodness of fit test—the goodness of fit measuring the null hypothesis that the model equates the saturated one (Agresti, 2002; Knoke and Burke, 1980). In other words, we perform a model selection as described in the chapter eight of Agresti (2002) or chapter three of Knoke and Burke (1980). In Knoke and Burke (1980), the procedure is applied in a cause–effect model as in our case. Following the procedure of model selection, we first rank the ten models by the significance of their goodness of fit tests and then we select the significant models with the simplest design. In our case, all the models have significance value below 0.05 for the two tests we considered (Log Likelihood and Pearson). Therefore, all the models can be used in the analysis. Table 10 ranks all the models, illustrating their significance. Table 10 indicates that the best two models have the design of the original model. In particular, they show no higher dependency than pair wise associations. In addition, Table 10 shows that the design of the original model is consistently similar in the ten models: all the designs have order-two terms, no higher term is included; all the variables describing the original model are involved in all the ten models; Group × DC appears nine times, Group × SAW always, DC × M appears eight times, the three correlations appear simultaneously in seven designs. Altogether, the selection method applied to the holdout re-sampling technique shows that the design of the original model can represent our data.

We can conclude that the proposed original design well represent our data.

Again following the procedure in (Agresti, 2002; Knoke and Burke, 1980) we derive the parameters of the original model and compute the odds ratios with respect to Group.

Odds ratios less than 1 denote a preference toward PP group, greater than 1 a preference toward NPP group. Cells with “1” indi-

Table 10
Goodness of fit tests. Boldface values indicate the best two models.

Best fit models over the 60%	Goodness of fit over the 40%	
	Likelihood ratio significance	Pearson χ^2 significance
(GROUP × DC, GROUP × SAW, GROUP × M, DC × M, JS)	**0.9441**	**0.942**
(GROUP × DC, GROUP × SAW, GROUP × M, DC × M, GROUP × JS)	**0.491**	**0.760**
(GROUP × DC, GROUP × SAW, DC × M, JS)	0.6484	0.3489
(GROUP × DC, GROUP × SAW, DC × M, JS)	0.6279	0.163
(GROUP × DC, GROUP × SAW, DC × M, JS)	0.5623	0.1037
(GROUP × DC, GROUP × SAW, GROUP × M, JS × M)	0.5589	0.4323
(GROUP × DC, GROUP × SAW, DC × M, JS)	0.4087	0.1002
(GROUP × SAW, GROUP × M, DC × M, JS)	0.3321	0.3621
(GROUP × DC, GROUP × SAW, GROUP × JS, DC × M)	0.2119	0.273
(GROUP × DC, GROUP × SAW, DC × M, JS)	0.1756	0.0186*

* Non-significant tests.

Table 11
Odd ratios with respect to Group.

SAW	JS	*M*	DC	Odds
1.00	1.00	1.00	1.00	0.21
1.00	1.00	2.00	1.00	0.57
2.00	1.00	1.00	1.00	0.85
1.00	1.00	1.00	2.00	0.94
1.00	2.00	1.00	1.00	1.00
2.00	1.00	2.00	1.00	2.26
1.00	1.00	2.00	2.00	2.49
1.00	2.00	2.00	1.00	2.90
2.00	1.00	1.00	2.00	3.74
2.00	2.00	1.00	1.00	4.35
1.00	2.00	1.00	2.00	4.79
2.00	1.00	2.00	2.00	9.91
2.00	2.00	2.00	1.00	11.54
1.00	2.00	2.00	2.00	12.70
2.00	2.00	1.00	2.00	19.06
2.00	2.00	2.00	2.00	50.56

cate positive values and "2" negative values of the variable. For example, the probability that developers are satisfied of their job (JS = 1), have a sustainable amount of work (SAW = 1), participate to well organize meetings (M = 1), and receive quickly communication of design changes (DC = 1), is higher in the PP group than in the NPP group (five times more as odds = 0.21). Similarly this happens when three out of four factors have positive answer (grey zone of the top of Table 11). The opposite situation, when all these factors simultaneously are not met (value = 2), has probability to happen much more (more than fifty times) in the NPP group than in the PP group. This also happens when three out of four factors has positive answer (light grey zone of the bottom in Table 10).

6. Results and discussion

Following the traditional research in IS, we have defined a cause–effect model of job satisfaction in collaborative environments (Fig. 2). Supported by the literature of models of job satisfaction and studies in the agile development, and inspired by the Hygiene factors of Herzberg and JCM, we have identified the causes of job satisfaction in "work sustainability," "communication," and "work environment" and a set of their sub-factors (Table 2 and Fig. 2). We have validated and refined the model with a retrospective study on a sample of one hundred eight respondents. We have adopted a fine grained control of the possible bias that in general affects retrospective studies. With multiple correlation tests we have refined the model as in Fig. 7. In particular, the final model does not present factors related to "Work Environment," as instead forecasted in the introductory book on XP of Kent Beck (2000). This result, in fact, does not surprise too much as the influence of work environment on job satisfaction is controversial in literature. In traditional work in IS, open spaces show a negative influence on job satisfaction as well as in some of the studies on PP where, for example, rules like "screen sharing" are too demanding (Evans and Johnson, 2000; Müller and Tichy, 2001; Brennan et al., 2002; Gallis et al., 2003). On the other hand, there is evidence that in the industrial settings the work environment adopted with PP increases job satisfaction (Mannaro et al., 2004).

Communication and work sustainability are instead the major causes of developers' job satisfaction in our model (Research Question #1 and Fig. 6). In particular, "Communication increases job satisfaction" supports existing findings on collaborative work (including agile development) for which group cohesion and employees' interaction are positive causes of job satisfaction (Hackman and Lawler, 1971; Domino et al., 2003; Acuña et al., 2009). We need to say that not all the types of communication we considered are relevant for job satisfaction, though. In our study, the type of communication that relates to job satisfaction pertains to communication among developers and with other departments through meetings well organized, (Table 7). This result reflects the collaborative nature of modern software development processes. On the other hand, our findings on the whole sample do not support that quick communication of design artefacts is a factor of job performance despite recent literature on PP (Melnik and Maurer, 2002). If we differentiate the analysis in the two groups of the sample (PP and non PP) though, quick communication of design changes is more relevant in the group using PP (Table 8) although it is still not a factor that increases job satisfaction (Table 9). This result tells us that communication of design changes represents part of the communication activities among units in collaborative settings – as it is correlated to communication with departments, (Table 7) –, is relevant in teams adopting PP (Beck, 2000), but is not a factor of job satisfaction in any case (with or without the use of PP) in that this finding does not support the subjective perceptions of pair programmers reported in (Melnik and Maurer, 2002).

With our research, we also support that "Work sustainability increases job satisfaction" in modern collaborative development processes (Fig. 7) in that sustainable workload, no stress, and high task significance positively influences job satisfaction as in (Dybå and Dingsøyr, 2008; Rajeswari and Anantharaman, 2003; Domino et al., 2003). The model in Fig. 7 does not include "balance between work and home" as in (Sonnentag et al., 1994), though. Given the correlation between this factor and "sustainable amount of work" (Table 7) respondents might have interpreted that balancing between work and home is a form of workload and as such they have not directly connected it to job satisfaction.

In addition, looking at Table 9, we must say that our findings do not support the statement "Work sustainability increases job satisfaction because of the use of Pair Programming," as in (Acuña et al., 2009) and (Highsmith, 2003) and for any of the sub-factors of work sustainability.

Overall, we have found that the use of Pair Programming does not vary substantially the design of the model of Fig. 7 (Research Question #3 and Table 9). In other words, the relation between factors and job satisfaction of the model does not change neither in number or intensity with use of PP, the only exceptions being "good communication between departments," and "meeting well organized" (namely, partial correlations in Table 9 do not include these factors anymore). We explain this with the fact that responsibility and ownership of work is shared in the pair. As a consequence, developers are more confident of their knowledge and the communication with other departments becomes easier increasing job satisfaction. In case of "meeting well organized," the result supports the beneficial effects on satisfaction of using stand up meetings and the continuous interactions of PP (Beck, 2000).

Finally, in step three, we have isolated those factors of Table 2 that are most significantly influenced by the use of PP (Table 8). Sustainable amount of work, good communication between developers, design changes quickly communicated, and meeting well organized are the factors most sensitive to the use of the practice (Research Question #2). In addition, the comparative analysis of the in the two sample groups performed with odd ratios (Table 11) reveals that the probability to have satisfied developers, that have a good communication with the designers, participate to well organized meetings, and work with a sustainable pace is five times more when using Pair Programming and fifty times less when all these statements are negated.

7. Conclusions

In our paper, we propose and validate a model of job satisfaction of developers working in the collaborative environment of modern software organizations. The model identifies two major causes of job satisfaction, communication and work sustainability. Our findings report that the use of Pair Programming does not modify the original model of job satisfaction except in the case of the communication between departments and the well organized meetings which we motivated with the higher confidence of developers in the knowledge they acquired with the continuous interactions and meetings of Pair Programming. In addition, our model does not support the statement that a specific arrangement of the work environment enhances job satisfaction as claimed in the introductory book on Extreme Programming (Beck, 2000).

What our analysis supports, instead, is that the combination of high job satisfaction, meeting well organized, good communication with the designers, and a sustainable amount of work is much likelier with the adoption of Pair Programming. This would have been an obvious conclusion as pair programmers are typically very motivated and enthusiastic developers, had we not controlled for factors extraneous to the simple use of the practice. Namely, we have found no difference in demographic characteristics and expectations from the work like motivation, perceptions, or personal credentials when using Pair Programming. Therefore our findings can be explained purely by the use of the practice.

This conclusion suggests the Pair Programming harmonizes satisfaction, communication, coordination, and sustainability of work in a collaborative setting. The managers that introduce this practice in a team are likely to get this combination of positive factors in their development processes.

8. Future research

Our results contribute to the research on developers' job satisfaction in two directions: (1) defining a cause–effect model of job satisfaction in the industrial context for modern software methods of developments and (2) studying the impact of a modern practice of development on the model variables and relations. In our model, we consider organizational and objective characteristics of work collaboration, but a wider set of characteristics of the collaborative work might be considered, for example including relational aspects as in (Goldstein and Rockart, 1984).

The impact of one or more practices of development on social and behavioural models is still an open issue. The empirical evidence of the effects of the simultaneous use of more practices will be matter of future research.

Appendix A.

As an example we describe the design of a saturated model in two categorical variables with contingency table $I \times J$ (Agresti, 2002; Lloyd, 1999). The cells of the contingency table include expected frequencies $\{\mu_{ij} = n\pi_{ij}\}$ with π_{ij} being the cell probabilities. The log-linear model is therefore

$$\log(\mu_{ij}) = \lambda + \lambda_i^x + \lambda_j^y + \lambda_{ij}^{xy}$$

where $\lambda = \log(n)$, $\lambda_i^x = \log(\pi_i)$ and $\lambda_i^y = \log(\pi_j)$, $\lambda_{ij}^{xy} = \log(\pi_{ij})$ and $\pi_i \cdot \pi_j$ are the marginal probabilities of the contingency table $I \times J$.

The terms λ_{ij}^{xy} express the association of the variables x and y at the i-th row and the j-th column. When the variables are statistical independent then $\mu_{ij} = n\pi_i\pi_j$ – n times the product of marginal expected probabilities – and the terms λ_{ij}^{xy} do not appear.

Through a MLE process carried out on a given dataset we can estimate all the parameters in the right side of the equation. At this point we are able to predict μ_{ij} for any value of i and j. In particular, we can compute the expression $\log(\mu_{ij}/\mu_{ik}) = \lambda_j^y - \lambda_k^y + \lambda_{ij}^{xy} - \lambda_{ik}^{xy}$ that is the difference in two values of y, j and k, for the same value of x, i. This expression is used to compute the odds ratio of a dichotomous variable. For example, if Z is a dichotomous variable with values 1 and 2, the odds ratio of Z with respect to the variable X for the value i is:

$$\text{Odds ratio of } Y = \log\left(\frac{Pr(Y=2;X=i)}{Pr(Y=1;X=i)}\right) = \log\left(\frac{\mu_{i2}}{\mu_{i1}}\right) = \lambda_2^y - \lambda_1^y + \lambda_{i2}^{xy} - \lambda_{i1}^{xy}$$

Appendix B. The core questionnaire for the two groups of the sample

Experience of the respondent

1. What is your experience with the application domain you are working in?

- More than 5 years
- 1 to 5 years
- Less than 1 year

2. How long have you been programming?

- More than 5 years
- 1 to 5 years
- Less than 1 year

3. How long have you been using the programming language that is used to develop the software you are working on?

- More than 5 years
- 1 to 5 years
- Less than 1 year

4. How long have you been programming using a non-PP process?

- More than 2 years
- 1 to 2 years
- Less than 1 year

B.1. 5. Practices of software development

5. When programming, do you do design reviews?
6. What approach do you use?

- Review design while creating design
- Review after completing the design
- Review during and after design
- Do not do a design review
- Other

7. When programming, do you do a code review?
8. What approach do you use?

- Review code while creating code
- Review code once coding is done, before compiling
- Review code once coding is done, after compiling
- Do not do a code review
- Other.

9. When programming, do you do unit testing?
10. What approach do you use?

- Tests are written before code
- Tests are written after code in order to fit it
- No unit testing is done
- Other

B.2. Job satisfaction

12. How do you feel about your job?

- Very satisfied
- Satisfied
- Unsatisfied
- Very unsatisfied

B.3. Perceived factors of job satisfaction

13. How much do you consider the following factors influence your job satisfaction? (rank how strongly you agree with each statement: H = very important, I = important, N = not important)

- Working on a good team with good team members
- The task I am assigned to
- The work environment
- The project I am assigned to
- Salary
- Being able to see the finished product and not only an intermediate version of it
- Communication with the management
- Communication with other employees

B.4. Work sustainability

14. Rank how strongly you agree with the following (SA = strongly agree, A = agree, D = disagree, SD = strongly disagree):

- I am very satisfied with my work
- My work is a source of worry and stress
- I have too much work and the tasks I am assigned to take up all my time
- I manage to balance between home and work
- My work has become too monotonous

B.5. Work environment

15. How do you consider your working environment? (Rank how strongly you agree with each statement VG = very good, G = good, S = satisfactory, NG = not good):

- Workspace and office layout (desk and computer space)
- Lighting
- Noise
- Heating
- Communication

16. Rank how strongly you agree with the following (SA = strongly agree, A = agree, D = disagree, SD = strongly disagree):

- Communication between departments is satisfactory
- Communication between developers is good
- Design changes are communicated quickly
- Meetings are well organized

References

Acuña, S.T., Gomez, M., Juristo, N., 2009. How do personality, team processes and task characteristics relate to job satisfaction and software quality? Information and Software Technology 51, 627–639.

Agresti, A., 2002. Categorical Data Analysis. John Wiley and Sons.

Allen, M.J., Yen, W.M., 2002. Introduction to Measurement Theory. Waveland Press, Long Grove, IL.

Arisholm, E., Gallis, H., Dyba, T., Sjøberg, D.I.K., 2007. Evaluating pair programming with respect to system complexity and programmer expertise. IEEE Transaction on Software Engineering 33 (2), 65–86.

Beck, K., 2000. eXtreme Programming Explained: Embrace Change. Addison Wesley.

Benjamini, Y., Hochberg, Y., 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. Journal of the Royal Statistical Society Series B 57, 289–300.

Benjamini, Y., Yekutieli, D., 2001. The control of the false discovery rate in multiple testing under dependency. Annals of Statistics 29 (4), 1165–1188.

Brennan, A., Chugh, J.S., Kline, T., 2002. Traditional versus open office design. A longitudinal field study. Environment and Behaviour 34 (3), 279–299.

Brooke, P.B., Price, J.L., 1989. The determinants of employee absenteeism: an empirical test of a causal model. Journal of Occupational Psychology 62, 1–9.

Bryant, S., Romero, P., du Boulay, B., 2006. The collaborative Nature of Pair Programming. In: Proceedings of the 7th International Conference on Extreme Programming and Agile Processes in Software engineering , (XP2006), Oulu, Finland, June, pp. 53–64.

Campbell, D., Stanley, J., 1963. Experimental and Quasi-experimental Designs for Research. Houghton Mifflin Company, Boston.

Canfora, G., Cimitile, A., Visaggio, C.A., 2005. Empirical study on the productivity of the pair programming. In: Baumeister, H., Marchesi, M., Holcombe, M. (Eds.), Extreme Programming and Agile Processes in Software Engineering, 6th International Conference, XP 2005, Sheffield, UK, June 18–23, 2005, Proceedings, Lecture Notes in Computer Science 3556 Springer. , pp. 92–99.

Cheney, P.H., Scarpello, V., 1986. Job satisfaction and IS research. Journal of Management Information Systems 2 (4), 21–36.

Cockburn, A., 2006. Agile Software Development: The Cooperative Game. Addison Wesley.

Cockburn, A., Williams, L., 2001. The cost and the benefits of pair programming. In: Succi, G., Marchesi, M. (Eds.), Extreme Programming Examined. Addison Wesley, pp. 223–248.

Domino, M.A., Collins, R.W., Hevner, A.R., Cohen, C.F., 2003. Conflict in collaborative software development. In: Proceedings of the 2003 SIGMIS Conference on Computer Personnel Research: Freedom in Philadelphia—Leveraging Differences and Diversity in the IT Workforce , pp. 44–51.

Dybå, T., Arisholm, E., Sjoberg, D.I.K., Hannay, J.E., Shull, F., 2007. On the effectiveness of pair programming. IEEE Software 24 (6), 12–15.

Dybå, T., Dingsøyr, T., 2008. Empirical studies of agile software development: a systematic review. Information and Software Technology 50, 833–859.

Emery, P., 2002. The Danger of eXtreme Programming. http://members.cox.net/cobbler/XPDangers.htm, May, 2002.

Evans, G.W., Johnson, D., 2000. Stress and open-offices noise. Journal of Applied Psychology 85 (5), 779–783.

Fishbein, M., Ajzen, I., 1975. Belief, Attitude, Intention, and Behavior: An Introduction to Theory and Research. Addison-Wesley, Reading, MA.

Frigon, L., Mathews, D., 1997. Practical Guide to Experimental Design. John Wiley and Sons.

Galina, P.B., Aleksandr, L.T., 2001. Job satisfaction and Satisfaction with life. Sociology Research 49 (4), 6–12.

Gallis, H., Arisholm, E., Dybå, T., 2003. An initial framework for research on pair programming. In: Proceedings of the 2003 International Symposium on Empirical Software Engineering (ISESE'03), pp. 132–142.

Gittins, R.G., Hope, S., 2001. A study of human solutions in eXtreme programming. In: Proceedings of the 13th Annual Workshop of the Psychology of Programming Interest Group, 17–20 April 2001, Bournemouth, UK, pp. 41–51.

Goldstein, D.K., Rockart, J.F., 1984. An examination of work-related correlates of job satisfaction in programmer/analysts. MIS Quarterly 8 (2), 103–115.

Hackman, J.R., Lawler, E.E., 1971. Employee reactions to job characteristics. Journal of Applied Psychology 55, 259–286.

Hackman, J.R., Oldham, G.R., 1980. Work Redesign. Addison-Wesley, Reading, MA.

Herzberg, F., Mausner, B., Snyderman, B.B., 1959. The Motivation to Work. John Wiley & Sons, New York.

Highsmith, J., 2003. Agile software development—why it is hot! In: Marchesi, M., Succi, G., Weels, D., Williams, L. (Eds.), Extreme Programming Perspectives. Addison Wesley, pp. 9–16.

Higman, J., Mackinnon, T., Moore, I., Pierce, D., 2003. Innovation and sustainability with gold cards. In: Marchesi, M., Succi, G., Weels, D., Williams, L. (Eds.), Extreme Programming Perspectives. Addison Wesley.

Humphrey, W., 2001. Comments on the eXtreme Programming, eXtreme Programming Pros and Cons: What Questions remain? IEEE Computer Society Dynabook, http://www.computer.org/seweb/dynabook//HumphreyCom.htm.

Igbaria, M., Parasuraman, S., Badawy, M.K., 1994. Work experiences. Job involvement, and quality of work life among information systems personnel. MIS Quarterly 18 (2), 175–201.

Janes, A., Russo, B., Succi, G., 2003. An empirical analysis on the discontinuous use of pair programming. In: Marchesi, M., Succi, G. (Eds.), Extreme Programming and Agile Processes in Software Engineering, Lecture Notes in Computer Science. Springer, pp. 205–214.

Judd, C.M., Smith, E.R., Kider, L.H., 1991. Research Methods in Social Relations, Sixth ed. Holt, Rinehart and Winston, Inc., Fort Worth, TX, USA.

Katira, N., Williams, L.A., Osborne, J., 2005. Towards increasing the compatibility of student pair programmers. In: Roman, G., Griswold, W., Nuseibeh, B. (Eds.), Proceedings of the 27th International Conference on Software Engineering. St. Louis, MO, USA, May 15–21, ICSE 2005, pp. 625–626.

Keller, R.T., 1983. Prediction absenteeism from prior absenteeism, attitudinal factors, non attitudinal factors. Journal of Applied Psychology 68, 536–540.

Knoke, D., Burke, P.J., 1980. Log-linear Models. Sage Publications, Beverly Hills, CA.

Kohavi, R., 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proceedings of the Fourteen International Joint Confer-

ence on Artificial Intelligence, vol. 2 , Montreal, Quebec, Canada August 20–25, pp. 1137–1143.
Lakhan, V.C., Lavalle, P.D., 2002. Use of loglinear models to assess factors influencing concern for the natural environment. Environmental Management 30 (1), 77–87.
Layman, L., Williams, L., Cunningham, L., 2004. Exploring extreme programming in context: an industrial case study. In: The Agile Development Conference (ADC'04) , Salt Lake City, UT, pp. 32–41.
Lloyd, C.J., 1999. Statistical Analysis of Categorical Data. John Wiley and Sons.
Locke, E.A., 1984. Job satisfaction. In: Gruneberg, M., Wall, T. (Eds.), Social Psychology and Organizational Behavior. John Wiley & Sons, New York, pp. 93–117.
Mannaro, K., Melis, M., Marchesi, M., 2004. Empirical analysis on the satisfaction of IT employees comparing XP practices with other software development methodologies. In: Eckstein, J., Baumeister, H. (Eds.), Proceedings of the fifth International Conference on eXtreme Programming and Agile Processes in Software Engineering (XP 2004), Garmisch-Partenkirchen, Germany, June 6–10, Lecture Notes in Computer Science, Springer.
Martin, T.N., Hunt, J.G., 1980. Social influence and intent to leave: a path-analytic process model. Personell Psychology 33, 505–528.
McDowell, C., Werner, L., Bullock, H.E., Fernald, J., 2006. Pair programming improves student retention. Confidence, and program quality. Communications of the ACM 49 (8), 90–95.
Melnik, G., Maurer, F., 2002. Perceptions of agile practices: a student survey. In: Proceedings, eXtreme Programming/Agile Universe 2002, Lecture Notes in Computer Science ,. Springer Verlag, pp. 241–250.
Melnik, G., Maurer, F., 2006. Comparative analysis of job satisfaction in agile and non-agile software development teams. In: Abrahamsson, P., Marchesi, M., Succi, G. (Eds.), Proceedings of the Third International Conference on eXtreme Programming and Agile Processes in Software Engineering (XP2006), Oulu Finland, June 2006, Lecture Notes in Computer Science, Springer, 32–42.
Müller, C.W., Price, J.L., 1990. Economic, psychological, and sociological determinants of voluntary turnover. Journal of Behavioral Economics 19, 321–335.
Müller, M.M., Tichy, W.F., 2001. Case study: extreme programming in a University environment. In: Proceedings International Conference on Software Engineering (ICSE) , pp. 537–544.
Nagy, M.S., 2002. Using a single item approach to measure facet job satisfaction. Journal of Occupational and Organizational Psychology 75 (1), 77–86.
Nawrocki, J., Wojciechowski, A., 2001. Experimental evaluation of pair programming. In: Proceedings of the 12th European Software Control and Metrics Conference, ESCOM 2001, 2–4 April 2001 , London, pp. 269–276.
Nicholson, N., 1977. Absence behavior and attendance motivation: a conceptual synthesis. Journal of Management Studies 14, 231–252.
Nosek, J.T., 1998. The case for collaborative programming. Communication of the ACM 41 (3), 105–108.
Price, J.L., Müller, C.W., 1986. Absenteeism and Turnover among Hospital Employees. JAI Press.
Rajeswari, K.S., Anantharaman, R.N., 2003. Development of an instrument to measure stress among software professionals: factor analytic study. In: Madvivalla, M., Trauth, E. (Eds.), Proceedings of the 2003 SIGMIS Conference on Computer personnel Research: Freedom in Philadelphia—Leveraging Differences and Diversity in the IT Workforce 2003. Philadelphia, Pennsylvania April 10–12, pp. 34–43.
Raymond, E.S., 1999. The Cathedral & the Bazaar. O'Reilly.
Scarpello, V., Campbell, J.P., 1983. Job satisfaction and the fit between individual needs and organizational rewards. Journal of Occupational Psychology 56, 315–328.
Scarpello, V., Vandenberg, R.V., 1992. Generalizing the importance of occupational and career views to job satisfaction attitudes, University of Georgia, College of Business Administration. Journal of Organizational Behavior 12 (2), 125–140.
Schuman, H., Presser, S., 1981. Questions and Answers in Attitude Surveys. Experiments on Question Form. Wording and Context. Academic Press, New York.
Schwaber, K., 2004. Agile Project Management with Scrum. Microsoft Press.
Sfetsos, P., Stamelos, I., Angelis, L., Deligiannis, I.S., 2006. Investigating the impact of personality types on communication and Collaboration-Viability in Pair Programming—an empirical study. In: Abrahamsson, P., Marchesi, M., Succi, G. (Eds.), Proceedings of the Third International Conference on eXtreme Programming and Agile Processes in Software Engineering (XP2006), Oulu Finland, June 2006, Lecture Notes in Computer Science. Springer, pp. 43–52.
Sonnentag, S., Brodbeck, F.C., Heinbokel, T., Stolte, W., 1994. Stressor burn-out relationship in software development teams. Journal of Occupational and Organizational Psychology 67, 327–341.
Soonhee, K., 2005. Factors affecting state government information technology employee turnover intentions. The American Review of Public Administration 35 (2), 137–156.
Stoeva, A.Z., Chiu, S.K., Greenhaus, J.H., 2002. Negative affectivity, role stress and work—family conflict. Journal of Vocational Behavior 60, 1–16.
Succi, G., Stefanovic, M., Smith, M., Huntrods, R., 2001. Design of an experiment for quantitative assessment of pair programming practices. In: Proceedings of the Second International Conference on eXtreme Programming and Agile Processes in Software Engineering (XP2001) , Villasimius, Italy, May, 2001, Springer Verlag, pp. 18–23.
Succi, G., Marchesi, M., Pedrycz, W., Williams, L.A., 2002. Preliminary analysis of the effect of pair programming on job satisfaction. In: Abrahamsson, P., Marchesi, M., Succi, G. (Eds.), Proceedings of the Third International Conference on eXtreme Programming and Agile Processes in Software Engineering (XP2002). Springer Verlag, pp. 212–215.
Tessem, B., Maurer, F., 2007. Job satisfaction and motivation in a large agile team. In: Concas, G., Damiani, E., Scotto, M., Succi, G. (Eds.), Proceedings of the eighth International Conference on Agile Processes in Software Engineering and Extreme Programming (XP 2007). Como, Italy, June, pp. 54–61.
Williams, L.A., 2003. Guest editor, special issue on eXtreme Programming. IEEE Software, 20, 3.
Williams, L.A., Kessler, R., Cunningham, W., Jeffries, R., 2000. Strengthening the case for pair-programming. IEEE Software 4, 19–25.

Witold Pedrycz is a Professor and Canada Research Chair (CRC) in the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Canada. He is also with the Systems Research Institute of the Polish Academy of Sciences, Warsaw, Poland. He is actively pursuing research in computational intelligence, human-centric computing, fuzzy modelling, knowledge discovery and data mining, fuzzy control including fuzzy controllers, pattern recognition, knowledge-based neural networks, relational computation, bioinformatics, and software engineering and system engineering.

Barbara Russo received her Ph.D. degree in Mathematics from the University of Trento, Italy, in 1996. She was visiting researcher at the May Plank Institut für Mathematik in Bonn (D) and at the University of Liverpool (UK). She is currently an Associate Professor with Tenure of the Faculty of Computer Science at the Free University of Bolzano-Bozen, Italy. Her current research interest focuses on various topics of software and system engineering specifically related to model and predict software and system behaviour with techniques of data mining in particular in the field of Open Source software and agile software development.

Giancarlo Succi is Professor with Tenure at the Free University of Bolzano-Bozen, Italy, where he directs the Centre for Applied Software Engineering. Before joining the Free University of Bolzano-Bozen, he has been Professor with Tenure at the University of Alberta, Edmonton, Alberta, Associate Professor at the University of Calgary, Alberta, and Assistant Professor at the University of Trento, Italy. The research interest of Giancarlo Succi involve multiple areas of software engineering, including open source development, agile methodologies, experimental software engineering, software engineering over the Internet, and software product lines and software reuse. Giancarlo Succi is a Fulbright Scholar.