Experimental Pair Programming: A Study Design And Preliminary Results

Marcel Valový
Department of Information Technologies
Prague University of Economics and Business
xvalm00@vse.cz

Abstract

In the context of facing a period in time where traditional work settings have been challenged, we require exploring new ways of increasing motivation in software teams. Pair programming is one of the agile techniques that might increase the motivation of software professionals and, as a result, also their productivity and creativity. This paper presents a work-in-progress design for conducting mixed-methods controlled experimental research on assessing, evaluating, and exploring the meaning behind the relations between personality, motivation, and programming roles.

***Keywords**:* pair programming, experimental design, software engineering, intrinsic motivation, big five.

1. Introduction

This paper presents an experimental mixed-methods research design for establishing and measuring the strength of links between the agile software development practice of pair programming and the motivation of software professionals. Why is this important? First, the topic of pair programming is now actively discussed in both scientific and industrial communities in relation to remote work settings. Secondly, traditional motivation techniques are now less effective but, more than ever, motivated and cooperating team members are crucial for any software project's success.

During the pandemic, software engineers writing computer programs in solitude might be suffering from the frustration of the triad of basic psychological needs identified by Ryan & Deci (2017): competency, relatedness, and autonomy. A study based on factor analysis of 2225 reports from 53 countries indicates the pandemic had put stress on the psychological needs and, in turn, significantly hurt developers' wellbeing and productivity (Ralph et al., 2020). Pair programming could be employed as a remedy to satisfy the needs and boost intrinsic motivation, but it is not guaranteed to be helpful. Based on the meta-analysis of previous reports, it possibly carries both positive and adverse effects on the motivation and performance of software developers (Hannay et al., 2009a). But what effects do its roles have *separately* and do they affect programmers of different *personalities* the same? That was *not* examined by any of the existing studies, and this paper sets to present an initial work-in-progress query framework for that, together with initial results, and discuss possible extensions to the initial framework.

The author proposes a nomological network to represent the constructs of interest in this experimental design, their observable manifestations, and the inter-relations between them. The core constructs are programming role (independent variable), intrinsic motivation (dependent variable), and personality (moderating variable). To perform statistical tests, motivation and personality must be operationalized into observable variables. The former will be modeled within the Self-determination framework (Ryan & Deci, 2017) and the latter within the Big Five model (Goldberg, 1993). Data collection ought to be executed in controlled experiments and semi-structured interviews.

Research problem:

How to effectively measure the effect of pair programming on motivation in software teams.

Research questions:

RQ1: Do distinct pair programming roles affect programmers' motivation differently?
RQ2: Can psychometric tests improve the assignment of pair programming roles?

The five initially tested hypotheses were drawn on the synthesis of knowledge in multiple disciplines, including psychology, organizational behavior, and software engineering, followingly:

H1: Programmers have distinct personalities.

H2: Distinct personality types prefer different pair programming roles.
H3: Openness positively moderates motivation in the pilot role.
H4: Extraversion and agreeableness are essential for a motivated navigator.
H5: Neuroticism and introversion are detrimental to both pilot and navigator roles.

2. Preliminary Experimental and Methodological Design

During the winter semester of 2021, the author created a methodological and experimental design and employed it in university classrooms. The context overview with a brief explanation of methods used follows first, then the preliminary experimental design is illuminated.

2.1 Context and methods

The proposed research questions revolve around the research problem of effectively increasing motivation using agile development techniques, such as pair programming. To answer them, we opted for an experimental, mixed-methods research strategy.

In the context of a 2nd-year IT undergraduate software engineering course, three rounds of a controlled experiment were carried out. Afterward, the quantitative data from the experiments were analyzed using contemporary statistical methods to establish empirical links between personality dimensions and software engineers' attitudes (Feldt et al., 2010; Graziotin et al., 2021). Additionally, we conducted semi-structured interviews with twelve of the experiment participants after the experiments finished and evaluated them using qualitative methods such as thematic analysis (Braun & Clarke, 2006), following the precise method of theme construction proposed by Vaismoradi et al. (2016).

2.2 Experimental design

The subjects were students who signed up for an advanced software engineering course at the undergraduate university level during the winter semester of the academic year 2021/22.

The author ran three laboratory sessions of 60-minutes net programming time each, during which the (N = 40) subjects took a break every 10 minutes to self-report their motivation with a seven-item standardized questionnaire “Intrinsic Motivation Inventory” (Deci & Ryan, 1982), rotate in pairs and receive a new task (yet being able to continue on the previous one because the tasks were continuous).

One group of subjects worked in pairs, and the subjects in another – the “control group” – worked alone. The partners of each pair were either designated “pilot” who controls the keyboard and codes, or “navigator” who conceptualizes the solution to the given task and looks for defects, with the subjects told to switch roles every 10 minutes.

The last session was without a control group. This effectively put each individual in three different conditions or “roles” (solo, PP-pilot, PP-navigator) for 6x10 minutes, yielding 6 motivation measures for each individual in each condition.

Each individual’s “preferred role” is then related to his or her personality. The preferred role is defined as the condition with the highest average reported motivation level. The subject’s personalities were measured with the Big Five personality test (Rammstedt et al., 2013) at the beginning of each session.

The subjects were instructed on how to pair-program during a 60-minute *pilot session* that preceded the three experimental sessions.

The subjects were working on predefined tasks in a static order. During the first session, the purpose of the tasks was to develop a contextual menu for the first semestral project, which is an adventure game in Java with a graphical user interface in JavaFX. In the second session, the tasks were about animating elements in the game. In the third session, participants developed the core of their team projects.

No external motivators were used, i.e., no credits were given for achieving correct solutions.

The task difficulty presented the main concern to the validity of the experimental design and that is why the first two sessions contained a “control group”. In each session, the motivational differences between each 10-minute round were compared in both the control (solo programming) and the test group, and statistical tests confirmed that there was, in fact, no relation between the tasks and self-reported intrinsic motivation. This is consistent with the findings of Vanhanen and Lassenius (2005).

It is also worth noting that the subjects were of various backgrounds and abilities, and this would carry a greater possible distortion on the results if performance were measured (Arisholm et al., 2007) as *performance = ability x motivation* (Latham, 2012). We diminished those effects by measuring intrinsic motivation which depends on autonomy, competence, and relatedness.

The effect of "pair jelling", i.e., relative improvement after the first task mentioned by Williams et al. (2006) was tested statistically and did not project itself into the motivation results.

Pairs were allocated randomly and irrespective of personality, similar to two existing experimental studies, one with 564 students and 90 % of pairs reporting compatibility (Katira et al., 2004) and another with 1350 students and 93 % reported compatibility (Williams et al., 2006).

3. Preliminary Results

The five preliminary hypotheses were confirmed and both preliminary RQ1 and RQ2 were answered positively, both by preliminary quantitative and qualitative results. The author would like to build on these results and extend the design. First, the quantitative results are presented, followed by the qualitative ones.

3.1 Quantitative results

Of 40 students, 2 were females and 38 were males. The students' software engineering experience ranged from a half to six years ($\bar{x} = 2.2$, $\sigma = 1.5$): 19 had up to one year of experience, 13 had more than one and up to three, and 8 had more than three and up to six years of software engineering experience.

Personality variables were mapped using clustering methods. The Dunn index metric (Bezdek & Pal, 1995) indicated the usage of a complete linkage method with the number of clusters set to three. The three centroids created by the hierarchical cluster analysis in RStudio (v4.0.5) are presented in Figure 1 with their respective means and standard deviations. The first cluster is characterized mainly by the predominant personality dimension "openness to experience" ($\mu = 8.29$, $\sigma = 1.21$). The second cluster is characterized by two dominant personality dimensions, "extraversion" ($\mu = 7.36$, $\sigma = 1.36$) and "agreeableness" ($\mu = 7.91$, $\sigma = 0.83$). The third cluster is characterized by the predominant personality dimension "neuroticism" ($\mu = 7.82$, $\sigma = 1.40$) and very low "extraversion" ($\mu = 3.55$, $\sigma = 1.04$).

		O	C	E	A	N
Cluster 1	η	8.29	6.53	6.35	6.12	5.94
	σ	1.21	1.66	1.58	1.54	1.43
Cluster 2	η	6.09	6.00	7.36	7.91	5.00
	σ	1.76	1.48	1.36	0.83	1.48
Cluster 3	η	6.73	5.82	3.55	5.91	7.82
	σ	1.68	1.72	1.04	1.14	1.40

Figure 1 - Cluster centroids characterized by their means and standard deviations

The relations between clusters and preferred roles were assigned by passing each cluster to the role by the maximum intercept and are displayed in Figure 2 with their respective counts. Cluster 1 prefers the Pilot (11 matches), cluster 2 the Navigator (6 matches), and cluster 3 the Solo (6 matches) role.

	Navigator	Pilot	Solo
Cluster 1	5	11	1
Cluster 2	6	3	2
Cluster 3	3	2	6

Figure 2 - Contingency table of clusters and their preferred roles

Various statistical tests – out of this paper's scope – confirmed the statistical counterparts of H1-5.

3.2 Qualitative results

The qualitative results, produced by coding the responses of twelve interviewees (*I#01-12*) and evaluating them using thematic analysis, can be briefly summarized followingly:

From the qualitative data set, a total of 68 codes with 184 occurrences and 8 themes were generated.

First, the majority of occurring codes (31 out of 68), with a number of 106 occurrences, were positive. The four overarching themes generated from them included the perceived psychological, pedagogical, and therapeutic effects and also the topics of reduced cognitive load and increased performance:

Activation of the "Hawthorne effect",
Separating the concerns and essential complexities of software engineering,
Improved performance,
Teaching and knowledge sharing.

Secondly, a large proportion of analyzed codes (21 out of 68) came under a neutral tone and captured statements about the relation between the agile development practice and personality factors and stressed the importance of rules:

Psychometric testing and personality's moderating effect,
Importance of rules.

Lastly, there were two negative overarching themes produced from a minority of 16 codes, where the standard argument was that it is difficult to work with introverts or when the partner is on a superior level and does not listen:

Attention deficit and focus deterioration,
Personality and competency differences.

4. Extending the Design

While the results provided some valuable insight, the author believes there is still much to be explored. For instance, the current design does not dwell in the space of various "pairing constellations". Also, some of the qualitative results are contrasting and could seem contradictory. On the contrary, they represent the opposite ends of the spectrum of programmers' attitudes. Therefore, all opinions are equally valid for implementing pair programming and enhancing the motivation of a software engineering team and must be considered. The current design, however, does not differentiate between the views of persons coming from the "openness", "extraversion and agreeableness", and "neuroticism" clusters. It also does not provide insight on whether the members of each cluster work equally well (and in what roles) when paired with members of different clusters, i.e. on the "pairing constellations".

Thus, the author would like to build on the initial design and extend it followingly:

- Introduce pairing constellations. Before the experiment rounds commence, the psychometrics would be assessed, and consequently, the participants will be assigned to 9 pair constellations: Pilot-Openness/Navigator-Openness, Pilot-Openness/Navigator-Agreeableness+Extraversion, Pilot-Openness/Navigator-Introversion+Neuroticism, etc., which would be executed in two sessions and then compared to one session of participants' motivation reports in solo settings.
- For the qualitative part of the research, representatives of each cluster should be interviewed until the point of saturation is reached for the study of each cluster's members' motivations, values, and attitudes, leading to cluster-specific and general overarching themes.
- Pairs should be assigned to match competency levels, i.e. to prevent the re-occurrence of the last identified theme "Personality and competency differences".

Several new hypotheses could be drawn, e.g.:

1. *Openness pilots paired with Extraversion-Agreeableness navigators exhibit the highest increase in intrinsic motivation when compared to their solo-work intrinsic motivation.*
2. *The positive effect of pairing on intrinsic motivation is lesser for homogeneous pairs, i.e., consisting of members from the same clusters, than for heterogeneous pairs.*
3. *Members of the Neuroticism cluster prefer to work with members of the Extraversion-Agreeableness cluster rather than with the Openness or Neuroticism cluster, in both roles.*

5. Discussion

The author presented a preliminary version with preliminary results and a concept of an extended version of the design for studying the effects of pair programming on the intrinsic motivation of software professionals using controlled experiments and expects the discussion to continue at the PPIG workshop. While the preliminary design has proven to be viable and the preliminary results provided novel insight into the psychological aspects of programming, the untapped space remains vast and awaits scientific exploration. Advanced inquiry frameworks, such as those presented by Feldt et al. (2010) and Graziotin et al. (2021) allowed for measuring the links in our proposed nomological structure in a way that was not common before. This challenges one of the conclusions of the previous seminal paper by Hannay et al. (2009b) that the moderating effect of personality on pair programming is rather insignificant.

Personality was confirmed to be a valid predictor of intrinsic motivation in software engineering. Software engineering managers could try to infer from the psychometrics of their team and act based on the findings of future studies using the extended experimental pair programming study design.

The possible threats to the external validity of studies using the presented design are cultural aspects and competency level. In different countries, the personality clusters of software professionals can differ. Thus, as Latham & Pinder (2005) note, all motivational theories and frameworks should take the context of national culture into account. The studies should therefore be replicated before being acted upon in contexts of different regions. Another threat is the competency level. The results may not be applicable to experts with more than the maximum observed number of six years of software engineering experience in our subjects.

Acknowledgments

This work was supported by an internal grant funding scheme (F4/34/2021) administered by the Prague University of Economics and Business.

References

Arisholm, E., Gallis, H., Dyba, T., & Sjoberg, D. I. (2007). Evaluating pair programming with respect to system complexity and programmer expertise. *IEEE Transactions on Software Engineering*, *33*(2), 65-86.

Bezdek, J. C., & Pal, N. R. (1995, November). Cluster validation with generalized Dunn's indices. In *Proceedings 1995 second New Zealand international two-stream conference on artificial neural networks and expert systems* (pp. 190-190). IEEE Computer Society.

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative research in psychology*, *3*(2), 77-101.

Deci, E. L., & Ryan, R. M. (1982). Intrinsic motivation inventory [measurement instrument].

Feldt, R., Angelis, L., Torkar, R., & Samuelsson, M. (2010). Links between the personalities, views and attitudes of software engineers. *Information and Software Technology*, *52*(6), 611-624.

Goldberg, L. R. (1993). The structure of phenotypic personality traits. *American psychologist*, *48*(1), 26.

Graziotin, D., Lenberg, P., Feldt, R., & Wagner, S. (2021). Psychometrics in behavioral software engineering: A methodological introduction with guidelines. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, *31*(1), 1-36.

Hannay, J. E., Dybå, T., Arisholm, E., & Sjøberg, D. I. (2009a). The effectiveness of pair programming: A meta-analysis. *Information and software technology*, *51*(7), 1110-1122.

Hannay, J. E., Arisholm, E., Engvik, H., & Sjoberg, D. I. (2009b). Effects of personality on pair programming. *IEEE Transactions on Software Engineering*, *36*(1), 61-80

Katira, N., Williams, L., Wiebe, E., Miller, C., Balik, S., & Gehringer, E. (2004, March). On understanding compatibility of student pair programmers. In *Proceedings of the 35th SIGCSE technical symposium on Computer science education* (pp. 7-11).

Latham, G. P., & Pinder, C. C. (2005). Work motivation theory and research at the dawn of the twenty-first century. *Annual review of psychology*, *56*(1), 485-516.

Latham, G. P. (2012). *Work motivation: History, theory, research, and practice*. Sage.

Ralph, P., Baltes, S., Adisaputri, G., Torkar, R., Kovalenko, V., Kalinowski, M., ... & Alkadhi, R. (2020). Pandemic programming. *Empirical Software Engineering*, *25*(6), 4927-4961.

Rammstedt, B., Kemper, C. J., Klein, M. C., Beierlein, C., & Kovaleva, A. (2013). A short scale for assessing the big five dimensions of personality: 10 item big five inventory (BFI-10). *methods, data, analyses*, *7*(2), 17.

Ryan, R. M., & Deci, E. L. (2017). *Self-determination theory: Basic psychological needs in motivation, development, and wellness*. Guilford Publications.

Vaismoradi, M., Jones, J., Turunen, H., & Snelgrove, S. (2016). Theme development in qualitative content analysis and thematic analysis.

Vanhanen, J., & Lassenius, C. (2005, November). Effects of pair programming at the development team level: an experiment. In *2005 International Symposium on Empirical Software Engineering, 2005.* (pp. 10-pp). IEEE.

Williams, L., Layman, L., Osborne, J., & Katira, N. (2006, July). Examining the compatibility of student pair programmers. In *AGILE 2006 (AGILE'06)* (pp. 10-pp). IEEE.