International Journal of Software Engineering and Its Applications
Vol. 9, No. 3 (2015), pp. 123-136
http://dx.doi.org/10.14257/ijseia.2015.9.3.13

The Impact of Agile Methodology on Software Team's Work-Related Well-Being

Mazni Omar[1] and Sharifah Lailee Syed Abdullah[2]

[1]*School of Computing, Universiti Utara Malaysia, Malaysia*
[2]*Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Malaysia*
[1]*mazni@uum.edu.my*, [2]*shlailee@perlis.uitm.edu.my*

Abstract

Agile methodology is people-oriented. However, little evidence demonstrates the methodology effectiveness on humanistic aspects. Work-related well-being is measured to what extent the agile methodology can give impact on anxiety, contentment, depression, and enthusiasm level among software engineering (SE) teams. This paper aims to investigate empirically the effect of agile methodology on software development team's work-related well-being. To achieve this goal, a comparison study was carried out in an academic setting. A quantitative approach using statistical analysis was used to investigate the effect. Results showed that agile does not significantly affect work-related well-being. Nonetheless, the team that is able to apply the agile practices as closely as possible experienced higher level of enthusiasm during software project. This study provides additional empirical data in software engineering research and practices specifically on human aspects. Further investigation needs to be carried out on the software projects with higher task complexity.

Keywords: *Agile methodology, Work-related well-being, Software engineering teams, Human aspects, Anxiety, Contentment, Depression, Enthusiasm*

1. Introduction

Software is developed for people by people. This common statement highlights the fact that the software development and usage is a human-centred activity. The software life cycle activity starts with interaction between managers, developers, and users in understanding the software project. This process requires communication between various parties to ensure project goals can be achieved successfully. However, even though, from the outset, the project is highly dependent on human factors, there is very little exploitation of these phenomena in the SE domain. The possible reason for this omission is that software development is a complex activity that requires an understanding of various aspects of management, social, and psychological disciplines. It is not well understood how people learn to solve problems, which leads to the undesired blame casting for technological faults when problems are not resolved. In reality, the problem is simply due to lack of concern with human issues. In contrast, human issues in organizational success are widely recognized, and thus studied, in management field, as it is recognized that human resources are the most valuable assets in an organization. Thus, by understanding human aspects in software engineering, organizations can plan effective ways to improve productivity and performance of software engineering teams.

An effective software methodology may increase project success (Keil, Tiwana, & Bush, 2002). Thus, guidelines for choosing software methodology that suits the specific circumstances can help improve working practices amongst software developers, which would, in turn, lead to quality software. Methodologies, such as Structured Systems Analysis and Design Method (SSADM) and rapid prototyping, are more focused on

ISSN: 1738-9984 IJSEIA
Copyright ⓒ 2015 SERSC

technical issues, such as the use of computer-aided software engineering (CASE) tools in meeting the needs of the system. Although using CASE tools can accelerate the development of the system, it still does not guarantee the success of a project. Thus, the emergence of agile methodology, which emphasizes the importance of human aspects of the software development process, is expected to resolve the issue.

Past studies has demonstrated that agile developers are found to be enthusiastic (Syed-Abdullah, Holcombe, & Gheorge, 2006) and relaxed in a competent environment (Sharp & Robinson, 2010). It is widely believed that, when the developers are happy, they will write better code and thus, would increase quality and productivity in software engineering activities (John *et al.*, 2005; Lenberg *et al.*, 2014). However, there is little empirical evidence that shows the effectiveness of agile methodology on human and social factors (Dingsøyr & Dybå & (2012), Dybå & Dingsøyr, 2008). Therefore, further research focusing on human factors in software engineering, especially in application of agile methodology, is strongly recommended, as it would help yield valuable empirical evidence of the importance of developers' characteristics and well-being and their effect on the quality of software produced. This evidence can provide guidelines to practitioners on the applicability of software methodology and thus aid them in applying the methodology more effectively in practice. However, in order to identify the most suitable methodology to apply and the most effective means of its implementation in practice, the key concepts of software development methodology.

In general, software development methodology comprises two approaches—plan-driven methods and agile methods (Boehm & Turner, 2004). The plan-driven methods, also known as heavyweight methodologies, consist of the Waterfall, Spiral, and Rational Unified Process (RUP). On the other hand, Extreme Programming (XP), Scrum, Feature Driven Development (FDD), Lean Development, Dynamic Systems Development Method (DSDM), and Crystal Methods underline the characteristics of agile methods. The plan-driven method is based on formal steps, namely requirements definitions, analysis, design, coding, testing, implementation, and maintenance phase. Each phase requires a software development team to produce detailed software documentation deemed necessary for development of quality software. The documentation is required for easier referencing and understanding of the development process, which is of particular importance in maintenance phase. Agile methodology, in contrast, employs a lightweight process in which communication plays a more important role than overly comprehensive documentation. This method focuses more on a people-oriented approach, which relies on tacit or interpersonal knowledge whilst developing software. The Agile methodology aims to provide better guidelines for software developers and the management to incrementally improve their software process, which would in turn meet the needs of clients faster and to higher degree.

2. Agile and work-related well-being

Work-related well-being is broadly defined as the experience of positive feeling, such as happiness, contentment, comfort, and enthusiasm in the workplace (Warr, 1990, 2007). Research in human resources and management has identified by four factors that affect employee well-being—job design, performance monitoring, human resource practices, and team leader support (Frenkel, Tam, Korczynski, & Shire, 1998; Holman, 2002; Knights & McCabe, 1998).

According to Holman (2002), job design, along with job control, has a positive association with employee well-being. Moreover, although the author posits that an employee in an organization can benefit from job monitoring, a high level of monitoring has a negative effect on wellbeing (Chalykoff & Kochan, 1989; Holman, 2002). Human resources practices and team leader support factors are designed to better reflect managerial aspects in workplace. High-level support from management with higher

 Copyright ⓒ 2015 SERSC

control of job design is a major factor in influencing employee well-being (Holman, 2002; Kular, Gatenby, Rees, Soane, & Truss, 2008). Research on organizational psychology has also shown that the levels of employee well-being positively affect individual's job satisfaction (Wright & Bonett, 2007).

People feel happy at work when their job characteristics match their own potentials (Warr, 2007). Past research has shown that agile practices could successfully motivate team members and increase their job satisfaction (Melnik & Maurer, 2006; Sharp & Robinson, 2008; Tessem & Maurer, 2007), as they are developed to suit people's needs. For example, by using story cards during planning game activity, small chunks of functionality are discussed frequently with client, allowing team members to retain their feeling of enthusiasm (Syed-Abdullah *et al.*, 2006). In addition, during pair programming activity, members are required to solve programming problems in pairs and frequently swap partners, both of which promote collaboration and a sense of project ownership amongst the team members, and thus increase their well-being. Agile job characteristics, given that they place emphasis on the value of continuous feedback and frequent release, are able to reduce depression amongst members. This is because the practices promote a shared environment, which encourages members to have a clear direction towards achieving project goals. Therefore, it is posited that agile teams will experience a higher level of well-being compared to non-agile teams.

To date, research on the effect of well-being on software development teams has been conducted by Syed-Abdullah *et al.*, (2006). This study reveals that agile methodology has a positive impact on the well-being of the developers in terms of four specific dimensions—job related anxiety, depression, contentment, and enthusiasm level. These results show that by applying XP methodology, software development teams experience higher enthusiasm level in the most dynamic project. However, as different participant backgrounds and culture may affect empirical results, these findings need to be further examined in order to confirm their validity and generalizability in a different setting.

3. Methodology

The aim of this research was to provide empirical evidence for the effectiveness of an agile methodology on human aspects, which are work-related well-being. Therefore, this study in this study, the independent variable refers to the causes of the study, which are the software methodologies. Agile methodology was used as a treatment for software methodology, whereas Formal methodology was used as a control variable. Prior academic achievements, personality types, and team personality diversity were considered predictor variables, as these variables are predetermined and cannot be controlled (Harris, 2002). Before starting the experiments, the participants already had specific personal characteristics; thus, the researcher could not assign the participants randomly to different variables. Moreover, the small sample sizes limited the ability to control for the participants' personality types and prior academic achievements.

The dependent variable, work-related well-being was used to measure the effectiveness of the treatment (use of Agile methodology). During the experiment, extraneous factors were identified to neutralize confounding variables that might affect experimental results. The extraneous factors identified in this study include the work experiences of the SE teams, project types, tools and technology used, and work environments. Thus, in order to avoid any bias or unwanted effects on this study, all these factors were kept unchanged and identical for the experimental and control groups.

The experiments used pre- and post-test experimental and control group designs to measure the true cause-and-effect relationships of introducing new treatment (Dimitrov & Rumrill, 2003). In this design, several steps were taken:

i. Selecting SE projects. These projects were proposed by the lecturer and university staff members, who acted as clients.
ii. Grouping the student participants into teams.
iii. Randomization. The teams were assigned randomly to several different projects and divided into experimental and control groups. At least two groups were allocated for each software project.
iv. Administering pre-tests for both groups. In this study, two questionnaires (the work-related well-being questionnaire (Warr, 1990) was used to measure the teams' well-being and positive affectivity in a particular week. The Warr questionnaire measures the work-related well-being levels among team members during the software development activity.
v. Performing the treatment for the experimental group using agile methodology.
vi. Administering the post-test for both groups. Both groups were given the same pre- and post-test questionnaire and were tested at the same time

Research ContexT

The data analysed as a part of this study was collected in academic setting. In the academic setting, university students were selected as participants to develop client-based projects in a team. A comparison study was carried to investigate the impact of agile methodology on work-related well-being.

Comparison Study

Participants

The 67 students that took part in the study were divided into 16 teams for the purpose of the comparison study. The teams were randomly assigned to experimental and control groups. The odd-numbered teams (8 teams) were assigned to the control group (using formal methodology), and the even-numbered teams (8 teams) were assigned to the experimental group (using agile methodology). All teams were required to develop a web-based application according to the clients' needs. Each team had four or five team members. The team membership remained constant throughout the study.

Project Allocation

In this study, project allocation was randomly assigned to the teams. This is because all group members had the same level of prior academic achievements, as verified by the fuzzy membership function (Sharifah-Lailee, Mazni, & Mohd, 2011). The project complexity level was often perceived as a threat to experimental validity, as it was deemed unfair to assign a complex project to teams with lower programming skills. The project lecturers thus took the initiative to negotiate the project scopes with the clients, which ensured that each team had an equal chance of successfully developing projects with the same complexity.

4. Data Collection and Analysis

In order to assess the effectiveness of an agile methodology on work-related well-being, questionnaire (Warr, 1990) was used to measure software engineering teams' emotional level. The first task was to assess empirically whether there are any differences of work-related well-being level between Agile teams and Formal teams. Thus, the following hypothesis was formulated:

 Copyright ⓒ 2015 SERSC

H1A: Agile teams will experience higher level of work-related well-being compared to Formal teams (Well-being$_{AG}$ > Well-being$_{FL}$)

Work-related well-being scales contain four dimensions—anxiety, contentment, depression, and enthusiasm dimensions. Thus, the **H1A** hypothesis was further detailed into four hypotheses according to the dimensions.

H1$_{1A}$: Agile teams will experience lower level of anxiety compared to Formal teams (Anxiety$_{AG}$ < Anxiety$_{FL}$)

H1$_{2A}$: Agile teams will experience higher level of contentment compared to Formal teams (Contentment$_{AG}$ > Contentment$_{FL}$)

H1$_{3A}$: Agile teams will experience lower level of depression compared to Formal teams (Depression$_{AG}$ < Depression$_{FL}$)

H1$_{4A}$: Agile teams will experience higher level of enthusiasm compared to Formal teams (Enthusiasm$_{AG}$ > Enthusiasm$_{FL}$)

Reading timeline as Table 1 was based on three specific points in time—before treatment, during treatment, and after treatment. In the academic setting, six data readings were collected at specific points in time because the participants enrolled in two semester's software project. The time was divided into two phases—SE1 and SE2—because the Software Engineering Project is a one-year course. In order to test the different level of well-being, each participant's level of well-being were measured six times. This was done to investigate the fluctuation of moods at different time during software development phases.

Table 1. Reading Timeline

Reading Timeline	Description
Phase 1: SE1	
Time 1 (T1)	The pre-test reading before any treatment was given to the participants
Time 2 (T2)	During the treatment - focus on analysis and design
Time 3 (T3)	The post-test reading before the first project presentation to the client and evaluator
Phase 2: SE2	
Time 4 (T4)	Follow up treatment
Time 5 (T5)	During the treatment - focus on coding, refactoring, reverse engineering and testing activities
Time 6 (T6)	The post-test reading before the final project presentation to the client and evaluator

To further investigate the relationships between the number of agile practices used by team members and work-related well-being level, the following hypothesis was outlined.

H2$_{A}$: The number of agile practices used is positively correlated with work-related well-being

The H2$_{A}$ hypothesis was further separated into four sub hypotheses according to the work-related well-being dimensions. These are:

H2$_{1A}$: The number of agile practices used is negatively correlated with anxiety level

H2$_{2A}$: The number of agile practices used is positively correlated with contentment level

$H2_{3A}$: The number of agile practices used is negatively correlated with depression level

$H2_{4A}$: The number of agile practices used is positively correlated with enthusiasm level

Independent-samples t-test was used to compare the total mean score for well-being variable before the treatment was given to both teams. The reliability of the questionnaire was acceptable with Cronbach's alpha, α = .81. The results shows **no significant difference** at **T1** between the Agile (M_1 = 39.35, SD_1 = 6.21) and the Formal (M_1 = 38.81, SD_1 = 7.08) teams, $t(64)$ = .33, p = .74. Moreover, as the magnitude of the differences in the means was very small (η^2 = .002), this showed that levels of work-related well-being for both teams were similar before the treatment was given.

Further analyses were conducted to investigate the impact of software methodology on work-related well-being during the remaining five time points (T2-T6). The work-related well-being scales showed a satisfactory internal consistency coefficient, with Cronbach's alpha of .88, .84, .89, .91 and .92 at corresponding time points T2 − T6. The **mixed between-within analysis of variance, ANOVA** was used to investigate the effects of software development methodology treatment on changes of mood of work-related well-being across the five time periods (after treatment). The findings suggest that both teams experienced similar mood pattern for work-related well-being. Thus, the fluctuation of the teams' feeling could be related to the different stages in the development phases. All teams were experienced low levels of well-being at T2, T3, and T5, which related to the progress in the software development stages. However, the well-being level increased towards the project completion (Figure 1).

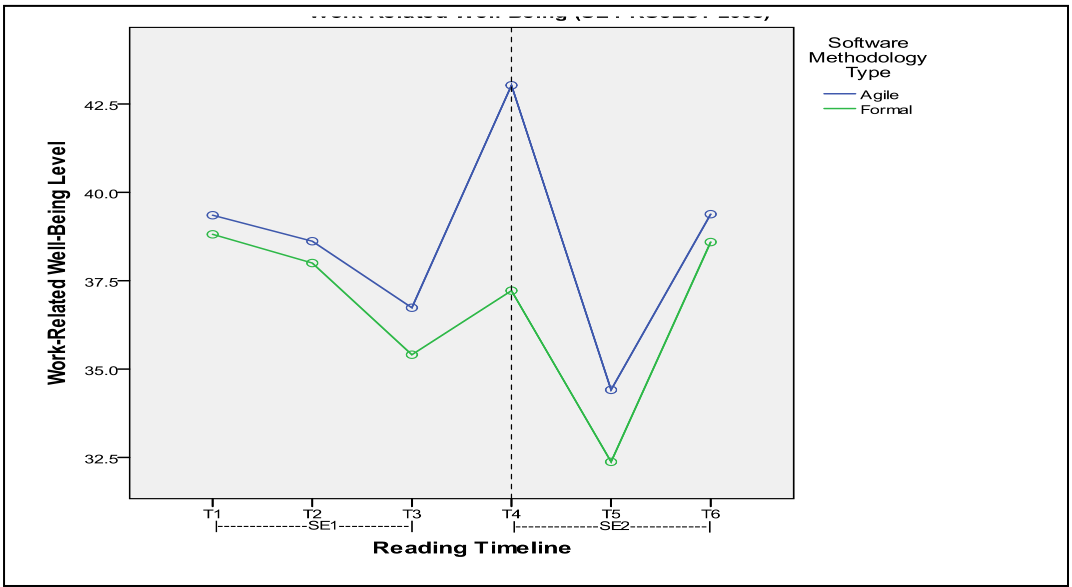

Figure 1. The Work-Related Well-Being Level for Agile and Formal Teams

The results show **significant difference** at **T4** between the Agile (M_4 = 43.03, SD_4 = 7.05) and the Formal (M_4 = 37.22, SD_4 = 7.42) teams, as indicated by $t(64)$ = 3.26, p = .014. The possible reason for this result is that the Agile teams have already gained the experience of applying agile practices during SE1. During this treatment (SE1) all Agile teams were rewarded higher marks compared to the Formal teams. Agile practices have encouraged them to work and develop software faster, unlike the Formal teams that were burdened with heavy documentation and therefore had limited time for coding during the

 Copyright ⓒ 2015 SERSC

first phase. Hence, it was not surprising that the Agile teams managed to complete 70% of the system's functionality.

There was significant effect for time, Wilks' Lambda = .52, $F(5,60) = 11.25$, $p < .0005$, multivariate $\eta^2 = .48$. In addition, interaction effect (Time × Type) shows statistically significant, with Wilks' Lambda = .82, $F(5,60) = 2.62$, $p = .03$, multivariate $\eta^2 = .18$. This shows that there were changes of moods across the six different time intervals for both teams.

Further analyses were carried out for each of the work-related well-being dimensions, i.e. anxiety, contentment, depression, and enthusiasm level. **Independent-samples t-test** was conducted to compare the mean score of the two software methodologies for each six intervals. Detailed results are discussed in the following sections.

a. Anxiety Level

Anxiety level refers to the feeling of tensed, worried, and anxious during the software project. Figure 2 illustrates the magnitude of the impact of software development methodologies on the anxiety level for the six intervals.

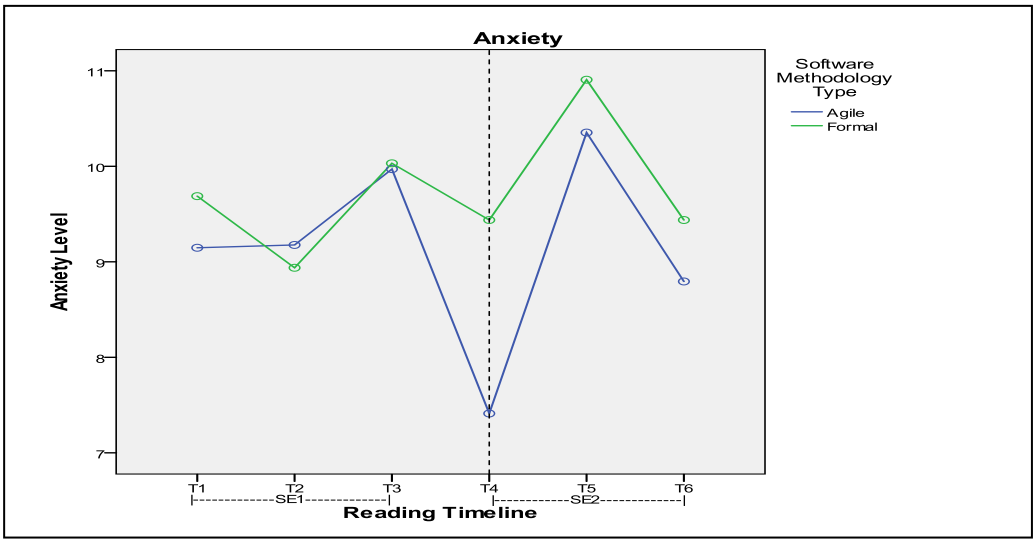

Figure 2. The Anxiety Level for Agile and Formal Teams

As can be seen from Figure 2, there was a **significant difference** between the two teams using different software methodologies at **T4**, with the Agile teams ($M_4 = 7.41$, $SD_4 = 2.43$) experiencing lower anxiety level compared to the Formal teams ($M_4 = 9.44$, $SD_4 = 2.68$), $t(64) = 3.23$, $p = .002$. The Agile teams experienced higher anxiety level compared to Formal teams at T2 because the Agile teams started to code their system whereas the Formal teams in the documentation stage. Lack of experience in applying agile practices and programming language explained the higher anxiety level at the beginning of project amongst agile team members. Nevertheless, as the Agile teams were rewarded with higher marks than were the Formal teams during SE1, the anxiety level of Agile teams decreased drastically at T4. However, both teams experienced similar pattern of anxiety level towards the end of software project. This is because both teams have gained experienced in understanding their project.

b. Contentment Level

Contentment level refers to the feeling of being calm, relaxed, and comfortable. In this study, the magnitude of the impact of software development methodologies on the contentment level for the six intervals is illustrated in Figure 3.

Copyright ⓒ 2015 SERSC

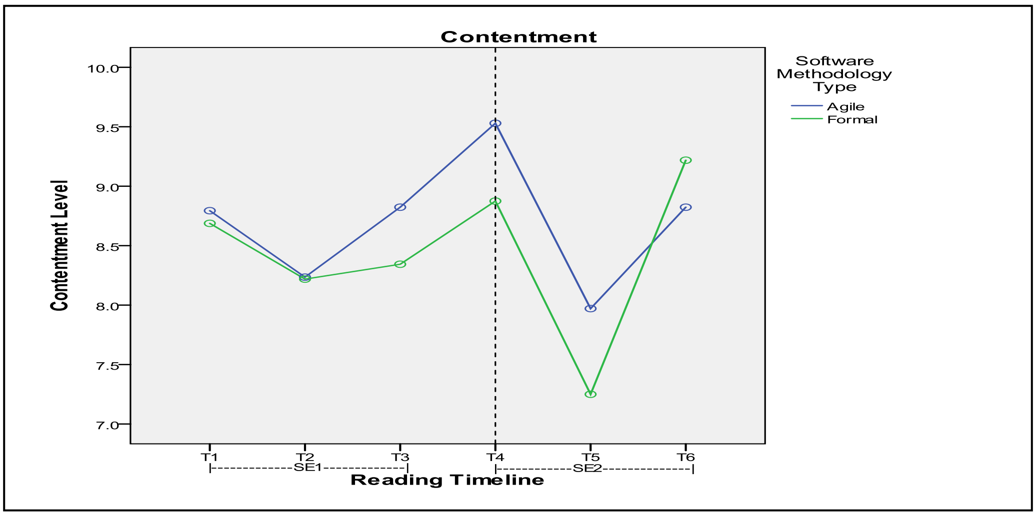

Figure 3. The Contentment Level for Agile and Formal Teams

It can be seen that there was **no significant difference** in the contentment level during the six intervals between the Agile and the Formal teams. Both teams experienced a decreased in the contentment level when the project was at coding phase (T2 and T5). This is because all members were requested to apply Java Servlet Pages (JSP), a new web-based technology that was difficult to apply by members. The Agile teams experienced slightly lower contentment level at T6 because lack of experience in applying agile practices. However, when the projects were ready to be delivered, all teams experienced higher contentment level resulting in **no significant difference** between the two software methodologies. Thus, the hypothesis **$H1_{2A}$ was rejected**. This finding supported earlier findings on the effect of agile methodology on contentment level (Syed-Abdullah et al., 2006).

c. Depression Level

Depression level measures the strength of the feelings of depression, misery, and gloom associated with the projects. Figure 4 illustrates the magnitude of the impact of software development methodologies on the depression level for the six intervals.

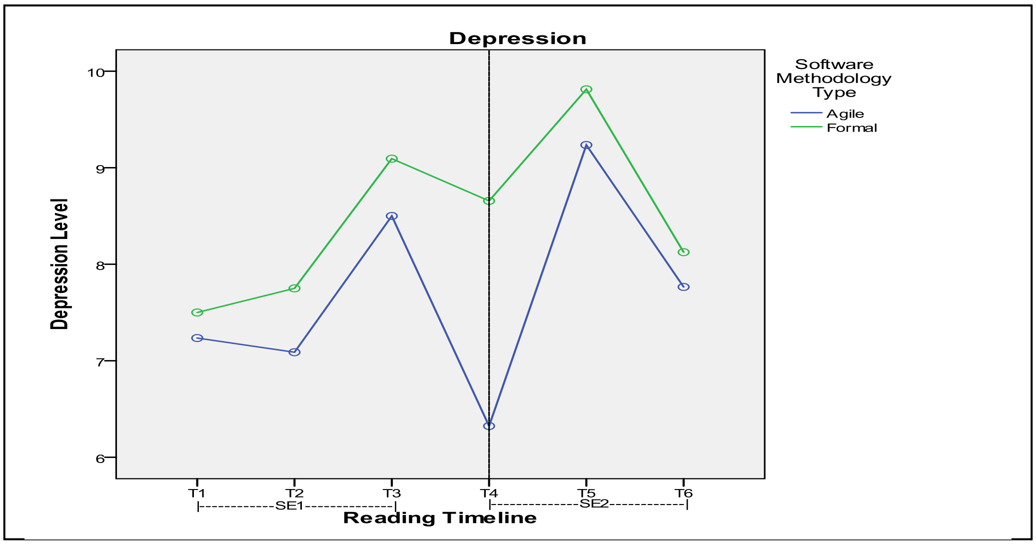

Figure 4. The Depression Level for Agile and Formal Teams

Copyright ⓒ 2015 SERSC

Both teams experienced similar pattern for depression level. However, there was a **significant difference** at **T4,** with the Agile teams (M_4 = 6.32, SD_4 = 2.73) experiencing lower depression level than the Formal teams (M_4 = 8.66, SD_4 = 2.53), $t(64) = 3.59$, $p = .001$. It appears that the perceived importance of the reward (high score awarded for their previous work) lessened the depression level amongst the Agile team members. Higher marks received during SE1 caused the Agile teams to feel less depressed because they believed that they were given more advantages than the Formal teams. The reason for this phenomenon is because the Agile teams were able to deliver the projects faster due to absence of the heavy documentation during SE1.

d. Enthusiasm Level

Enthusiasm refers to the feeling of being motivated, enthusiastic, and optimistic towards the project. Figure 5 illustrates the magnitude of the impact of software development methodologies on the enthusiasm level for the six intervals.

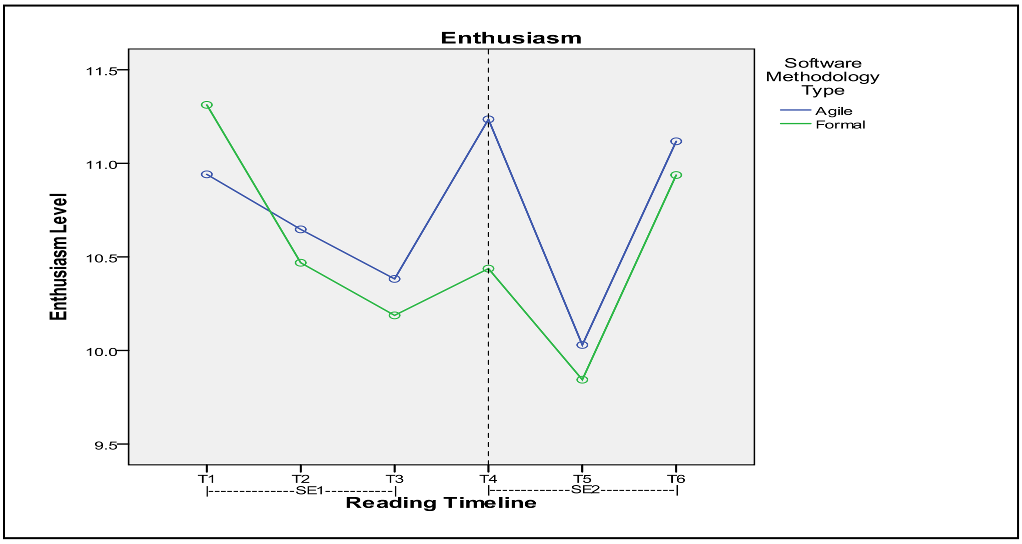

Figure 5. The Enthusiasm Level for Agile and Formal Teams

According to Figure 5, both teams experienced similar pattern of enthusiasm level, indicating that when there was **no significant difference** at all six intervals between the Agile and the Formal teams. Although at T1, the enthusiasm level of Formal teams were higher than Agile teams, the difference was not significant. However, the enthusiasm levels reported by the members of the Agile teams were higher at T4 because the teams received higher marks compared to the Formal teams. This reason had motivated the teams, increasing their enthusiasm levels. However, the level of enthusiasm started to fluctuate towards the end of the project. Therefore, as the comparison of enthusiasm level between the two methodologies showed **no significant difference**, the hypothesis **$H1_{4A}$ was rejected**. This finding failed to prove that the agile methodology could cause higher level of enthusiasm amongst the Agile teams compared to Formal teams.

e. Implications of Software Methodology on Work-Related Well-Being

Overall, results show that there were **no significant differences** for the four dimensions: anxiety, contentment, depression, and enthusiasm at the end of project. The experiment revealed that there was no significant effect of software development methodology on work-related well-being. As a summary; the results of hypotheses testing are exhibited in Table 2.

Copyright ⓒ 2015 SERSC

Table 2. Summary of the Impact on Work-Related Well-Being

Hypothesis Number	Hypothesis/ Sub Hypothesis	Empirical Results
$H1_A$	**Agile teams will experience higher level of work-related well-being compared to Formal teams ($Well\text{-}being_{AG} > Well\text{-}being_{FL}$)**	**Reject** However, the difference was significant at T4, which revealed that the Agile teams experienced higher level of well-being compared to the Formal teams.
$H1_{1A}$:	Agile teams will experience lower level of anxiety compared to Formal teams ($Anxiety_{AG} < Anxiety_{FL}$)	**Reject** However, a significant difference was found at T4, which revealed that the Agile teams experienced lower level of anxiety compared to the Formal teams.
$H1_{2A}$:	Agile teams will experience higher level of contentment compared to Formal teams ($Contentment_{AG} > Contentment_{FL}$)	**Reject**
$H1_{3A}$:	Agile teams will experience lower level of depression compared to Formal teams ($Depression_{AG} < Depression_{FL}$)	**Reject** However, the result at T4 showed significant difference, which revealed that the Agile teams experienced lower level of depression compared to the Formal teams.
$H1_{4A}$:	Agile teams will experience higher level of enthusiasm compared to Formal teams ($Enthusiasm_{AG} > Enthusiasm_{FL}$)	**Reject**

f. Number of Agile Practices Used and Work-Related Well-Being

Spearman's Rank Order Correlation was used to calculate the strength of the relationship between the number of agile practices and work-related well-being level. The result shows significant relationships for all the hypotheses tested, except for hypothesis $H2_{2A}$ that indicates **no significant relationship** between the number of agile practices used and the contentment level. This finding may be explained by noting that, in this study, the Agile teams needed to firstly become familiar with new agile practices. In contrast the Formal teams were well acquainted with the techniques they were supposed to use during the project. As a result of lack of familiarity, the Agile teams found it difficult to achieve contentment during software development Nevertheless, the results for hypotheses $H2_A$, $H2_{1A,}$ $H2_{3A,}$ and $H2_{3A}$ as found in Table 3 supported earlier findings (Syed-Abdullah *et al.*, 2006).

Table 3. Results of Relationships between Number of Agile Practices Used and Work-Related Well-Being Level

Hypothesis Number	Hypothesis/ Sub Hypothesis	Empirical Results
$H2_A$	**The number of agile practices used is positively correlated with work-related well-being**	**Accept** (r (34) = .58, p = .00)
$H2_{1A}$:	The number of agile practices used is negatively correlated with anxiety level	**Accept** (r (34) = - .63, p = .00)
$H2_{2A}$:	The number of agile practices used is positively correlated with contentment level	**Reject** (r (34) = .16, p = .38)
$H2_{3A}$:	The number of agile practices used is negatively correlated with depression level	**Accept** (r (34) = - .62, p = .00)
$H2_{4A}$:	The number of agile practices used is positively correlated with enthusiasm level	**Accept** (r (34) = .43, p = .01)

Copyright ⓒ 2015 SERSC

5. Discussion of Findings

In this study, similar patterns of the well-being levels existed for the Agile and Formal teams. It was noted that neither of the teams had any prior experience using Java Servlet Pages (JSP) technology. Thus, in this case, feeling of anxiousness was experienced by the members of both Agile and Formal teams. This study deduced that prior knowledge is very important in ensuring that the teams can easily and comfortably develop software within a specified time frame. Past researches have shown that the lack of knowledge is associated with higher anxiety level amongst team members (Armour, 2006; Axtell *et al.*, 2002; Salanova, Llorens, Cifre, Martínez, & Schaufeli, 2003).

Extrinsic motivators, such as rewards (grades), may mediate the level of well-being. Grade achievements often influence the members to appreciate all training programs. Such motivation enables the members to react positively and thus perceive the benefit of training program more positively. However, the opposite is also possible, in which when the grade is lower than the expected, the motivation might be even lower than if no grading was performed. Researches in human resources have shown that reward system affects the level of enthusiasm (Fernandez & Moldogaziev, 2011; Kirkman & Rosen, 1999).

Agile practices also promote knowledge sharing through pair programming, continuous integration, and frequent release. These practices are designed to minimize time pressure and lessen workload on the teams, thus reducing the feeling of depression during software development activities. In addition, knowledge disseminated during software development also serves to reduce the anxiety level of the software teams.

Besides, the application of agile practices allows the teams to develop high momentum, thus completing software more quickly. Pair programming and continuous integration are the key ingredients in encouraging team members to be more cooperative and thus more enthusiastic about their work. When members are engaged in activities that add value to the teams, they are able to achieve project goals, which results in increasing their well-being levels (Lent & Brown, 2008; Laanti, 2013). This finding suggests that collective work environment through the application of agile practices can increase level of enthusiasm amongst team members.

6. Conclusion and Recommendation

This study adds a new dimension regarding the effect of agile practices on team's work-related well-being. The findings showed that the use of agile methodology does not significantly affect work-related well-being. However, agile practices, such as pair programming, continuous integration, and frequent release, are able to induce teams to work closely and experience higher well-being. Thus, this study provides additional evidence on the advantages of agile methodology in software development.

It is suggested that future researchers might choose to extend this work by examining moderator variables such as leadership and motivation, which could potentially mediate the teams' psychological aspects and performance in software development. In addition, most empirical studies involved small-scale software projects which has low complexity tasks. Therefore, future work should focus on software projects with higher task complexity. This arrangement will be able to further determine the effect of task difficulty on teams' psychological well-being. Thus, the studies would provide additional empirical data which are relevant to software engineering research and practices.

Copyright ⓒ 2015 SERSC

References

[1] P. G. Armour, The learning edge. Commun. ACM. 49(6), 19-22 **(2006)**.
[2] C. Axtell, T. Wall, C. Stride, K. Pepper, C. Clegg, P. Gardner, Familiarity breeds content: The impact of exposure to change on employee openness and well-being. Journal of Occupational and Organizational Psychology. 75, 217-231 **(2002)**.
[3] B. Boehm and R. Turner, Balancing agility and discipline: A guide for the perplexed. Addison Wesley, USA **(2004)**.
[4] J. Chalykoff and T. A. Kochan, Computer-aided monitoring: Its influence on employee job satisfaction and turnover. Personnel Psychology. 42(4), 807-834 **(1989)**.
[5] T. Dingsøyr and T. Dybå, Team effectiveness in software development: Human and cooperative aspects in team effectiveness models and priorities for future studies. Proceedings of the 5th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), **(2012)** June 2; Zurich, Switzerland.
[6] D. M. Dimitrov and P. D. Rumrill, Pretest-posttest designs and measurement of change. Work. 20(2), 159-165 **(2003)**.
[7] T. Dybå and T. Dingsøyr, Empirical studies of agile software development: A systematic review. Journal Information & Software Technology. 50(9-10), 833-859 **(2008)**
[8] S. Fernandez and T. Moldogaziev, Empowering public sector employees to improve performance: Does it work? The American Review of Public Administration. 41(1), 23-47 **(2011)**.
[9] S. J. Frenkel, M. Tam, M. Korczynski & K. Shire, Beyond bureaucracy, Work organization in call centres. International Journal of Human Resource Management. 9(6), 957-979 **(1998)**.
[10] P. Harris, Designing and reporting experiments in psychology (2nd ed.). Open University Press, Buckingham **(2002)**.
[11] D. Holman, Employee wellbeing in call centres. Human Resource Management Journal. 12(4), 35-50 **(2002)**.
[12] M. John , F. Maurer and B. Tessem, Human and social factors of software engineering-workshop summary. SIGSOFT Software Engineering Notes. 30(4), 1-6 **(2005)**.
[13] M. Keil, A. Tiwana and A. Bush, Reconciling user and project manager perceptions of it project risk: A delphi study. Information Systems Journal. 12(2), 10-119 **(2002)**.
[14] B. L. Kirkman and B. Rosen, Beyond self-management: Antecedents and consequences of team empowerment. The Academy of Management Journal. 42(1), 58-74 **(1999)**.
[15] D. Knights and D. McCabe, What happens when the phone goes wild?': Staff, stress and spaces for escape in a bpr telephone banking work regime. Journal of Management Studies. 35(2), 163-194 **(1998)**.
[16] S. Kular, M. Gatenby, C. Rees, E. Soane and K. Truss, Employee engagement: A literature review. Kingston Business School, Surrey **(2008)**.
[17] M. Laanti, Agile and Wellbeing - Stress, Empowerment, and Performance in Scrum and Kanban Teams. Proceedings of the 47th Hawaii International Conference on System Sciences, **(2013)** January 7-10; Hawaii, USA.
[18] R. W. Lent and S. D. Brown, Social cognitive career theory and subjective well-being in the context of work. Journal of Career Assessment. 16(1), 6-21 **(2008)**.
[19] P. Lenberg, R. Feldt and L. G. Wallgren, Towards a Behavioral Software Engineering. Proceedings of the 7th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), **(2014**) June 2-3; Hyderabad, India.
[20] G. Melnik and F. Maurer, Comparative analysis of job satisfaction in agile and non-agile software development teams in Lecture notes in computer science, Edited P. Abrahamsson, M. Marchesi & G. Succi, Springer-Verlag, Berlin-Heidelberg **(2006)**, Vol. 4044, pp. 32-42.
[21] M. Salanova, S. Llorens, E. Cifre, I. M. Martínez and W. B. Schaufeli, Perceived collective efficacy, subjective well-being and task performance among electronic work groups an experimental study. Small Group Research. 34(1), 43-73 **(2003)**.
[22] S. A. Sharifah-Lailee, O. Mazni and F. I. M. I. Mohd, Team achievements equality using fuzzy rule-based technique. World Applied Sciences Journal. 15 (3), 359-363 **(2011)**
[23] H. Sharp and H. Robinson, Collaboration and co-ordination in mature extreme programming teams. International Journal of Human Computer Studies. 66(7), 506-518 **(2008)**
[24] H. Sharp and H. Robinson, Three 'c' of agile practice: Colloboration, co-ordination and communication in Agile software development: Current research and future directions, Edited T. Dingsøyr, T. Dybå & N. B. Moe, Springer-Verlag, London **(2010**), pp. 61-85.
[25] S. L. Syed-Abdullah, M. Holcombe and M. Gheorge, The impact of an agile methodology on the well being of development teams. Empirical Software Engineering. 11(1), 143-167 **(2006)**
[26] B. Tessem and F. Maurer, Job satisfaction and motivation in a large agile team. Proceedings of the 8th international conference on Agile processes in software engineering and extreme programming, **(2007)** June 18-22; Como, Italy.
[27] P. Warr, The measurement of well-being and other aspects of mental health. Journal of Occupational Psychology. 63, 193-210 **(1990)**.
[28] P. Warr, Work, happiness, and unhappiness. Lawrence Erlbaum Associates, Inc., New Jersey **(2007)**.

 Copyright ⓒ 2015 SERSC

Authors

Mazni Omar is a senior lecturer at the School of Computing, College of Arts and Sciences, Universiti Utara Malaysia. She holds PhD in Information Technology and Quantitative Sciences from Universiti Teknologi MARA, Malaysia. Her main research focuses are agile software development, empirical software engineering, software quality and data mining.

Sharifah Lailee Syed Abdullah is an Associate Professor at the School of Computer and Mathematical Sciences, Universiti Teknologi MARA. She holds PhD in Software Engineering from the University of Sheffield, United Kingdom (UK). Her main research focuses are empirical software engineering, data mining and intelligence agent.

Copyright ⓒ 2015 SERSC

Copyright ⓒ 2015 SERSC