An Empirical Analysis on the Effects of Agile practices on Motivation and Work Performance of Software Developers

Sungkeun Kim1, Soonsam Hwang2, Sehee Song3

Summary

‘Agile’ is drawing our attention as a new trend of software development methodology. With Agile methodology, developers can flexibly respond to changes in order to manage uncertainty and complexity of IT projects. Recent case studies are showing that international enterprises have adopted Agile methodology in order to improve work performance such as productivity and quality. Although further empirical researches as to what factors can improve performance have not been progressed yet, previous studies emphasized the developers’ motivation played an important role in producing good outputs in software development. Agile methodology is also known to be adequate to enhance both motivation and work performance of developers because the human factor is considered being an important component in it.

In this paper, we empirically analyzed by Multiple Regression the effects of various practices, which are essential for conducting Agile methodology, on the Motivating Potential Score(i.e. MPS) and on the work performance in development jobs. We first examined the Agile practices that are widely used for Agile developers in Korea, and divided them into management practices and development practices. As a result, it appeared that Agile practices exerted influences on the MPS of developers. While the MPS had positive effects on the work performance such as the development productivity and the system quality, it did not affect the project duration significantly. In our research, we will discuss practical significance based on such empirical results.

Keywords: Agile methodology, Agile practices, Motivating Potential Score(MPS), Work Performance, Multiple Regression

I. Introduction

It is not easy to successfully carry out an IT project. According to The Standish Group(2007), an average of 35 percents of IT projects throughout the world were brought to a successful finish with the requirements met within estimated time and cost. Anselmo & Ledgard(2003) also pointed out the bigger the
scale of a project gets, the more chances does the risk of exceeding planned time and cost exist. It is because a large scale of information system increases complexity and makes accurate estimation of required time and cost more difficult.

Schwaber (1996) asserted that a new software development methodology that is ideal to cope with changes is in need in order to manage uncertainty and complexity of IT projects. So far, a number of studies for finding such alternatives have been conducted and Agile methodology is coming into the spotlight for its effectiveness in continuous adoption of changes via repetitive yet incremental development.

In recent days, many case studies are showing that international enterprises which adopted Agile methodology have experienced improvements in work performance such as productivity and quality (Maurer & Martel, 2002; Muller & Padberg, 2002; Ilieva et al., 2004).

However, despite the positive results of adopting Agile methodology, such empirical studies as to what factors cause motivation to make impacts on the work performance have yet to be done.

In this study, we will examine the factors that affect the motivation of Agile developers and analyze them empirically.

Agile practices are detailed guidelines and yet the best practices which have been proven to be effective. Usage levels of practices differ by project teams and a variety of practices can be used in different circumstances. Therefore, a project that is driven by Agile practices may have a large influence on developers’ motivation and the work performance depending on how those practices are carried out.

We believe it will benefit effective adoption of Agile methodology that we present the usage level of what particular Agile practices are encouraged to be used in order to enhance MPS of software developers.

Software development typically requires intellectual labor and a developer’s performance largely depends on one’s capability.

Demarco & Lister (1999) asserted that the work environment to enhance one’s motivation is more important than technical factors such as development languages or tools for intellectual workers like software developers to improve their work performance.

In this study, we will verify the effects of Agile developers’ motivation on work performance such as productivity, quality and project duration based on the results of existing empirical studies.

The overall structure of this paper is as follows: The chapter II summarizes theoretical backgrounds of this study. The chapter III introduces a research model to evaluate the effects of Agile practices onto developers’ motivation and work performance. In the chapter IV, we verified the hypotheses that we formed by means of the research model we presented in the chapter III. Lastly in the chapter V, we suggested the significance and the limitation of this study as well as the direction in the future.

II. Theoretical Backgrounds

2.1 Agile Methodology and Practices

Agile methodology was disclosed to public by Agile Manifesto in 2001. The term ‘Agile’ encapsulates diverse development methodologies that have the characteristics of so-called ‘light weight methodology’. Typical Agile methodologies consist of XP (Extreme Programming), Scrum, Agile Unified Process,
Feature Driven Development (Cohn, 2003). In Korea, mainly web portal companies and game development companies have adopted and applied Agile methodologies to their processes. To examine the current status of Agile methodologies applied in Korea, we had interviews with engineers and project managers who had utilized Agile in their actual work and then analyzed the outcomes.

The result of the interview indicated three kinds of Agile methodologies were being used: XP, Scrum, and the compound methodology of XP + Scrum. Also it appeared that the most common Agile practices that developers used were derived from either XP or Scrum methodology.

We have classified a total of the 14 common practices stood out in the interview into two categories based on their purposes and characteristics. [Table 1] summarizes the seven items related with development and the rest of seven items related with management. In the table, it appears that XP is development-oriented whereas Scrum is management-oriented. Four items fall into the category of common practices which exist both in XP and Scrum.

2.2 Previous Studies in Agile Methodology

Lately, international enterprises are increasingly applying Agile methodology. According to Forrester Research in 2007, it was said that 25% of enterprises in the U.S. and Europe had adopted Agile methodology (Schwaber et al., 2008). As more and more enterprises are gradually adopting Agile methodology, many empirical studies have been introduced about it as well.

The empirical studies of Agile methodology have indicated Agile developers are more motivated than when they use other methodologies (Mannaro et al., 2004; Melnik & Maurer, 2006; Mann & Maurer, 2005; Ilieva et al., 2004).

Mannaro et al. (2004) stated the usage of Agile methodology improves developers’ motivation and it would eventually improve their work performance.

Thus, Agile methodology has been known to exert positive influences on the work performance during development. In particular, it directly affects productivity (Maurer & Martel, 2002; Muller & Padberg, 2002; Erdogmus & Williams, 2003; Ilieva et al., 2004).

Maurer & Martel (2002) and Muller & Padberg (2002) insisted the usage of Agile methodology improves productivity in development. Erdogmus & Williams (2003) mentioned it is a main cause to improve
productivity, too. Ilieva et al.(2004) mentioned Agile methodology had not only increased productivity but also reduced project overruns and defect rates.

Nonetheless, all the above studies merely verified the motivational improvement of by using Agile methodology. Empirical verification as to what factors make impacts on raising motivation has yet to be done.

2.3 Motivation and Work Performance of Software Developers

Previous studies about work performance in software development are showing that a developer’s motivation has the direct relationship with performance improvement(Baddoo et al., 2006; Procaccino et al., 2005).

Baddoo et al.(2006) stated in his case study that the motivation is an important determinant which raises performance in the organization. Procaccino et al.(2005) also insisted giving motivation to developers is important to satisfy user requirements and to accomplish successful projects.

Based on the study results that a developer’s motivation affects performance improvement, there have been a number of studies about the relationship between productivity and motivation (Kym & Park, 1992; Darcy & Ma, 2005).

In addition to the empirical studies about the relationship between motivation and work performance, there have been theoretical studies to supplement such empirical evidences.

There are three most widely used theoretical models among many of the studies about developers’ motivation. They are Maslow’s ‘five level hierarchy of needs’, Herzberg’s ‘dual factor theory’ and Hackman & Oldham’s ‘job characteristic model’ (Brenda L. & Hy S., 2001). Of them all, job characteristic model is the one that were the most commonly used for the studies from 1980 to 2006.

Previous studies have proven that the ultimate determinant that influences a developer’s motivation is the job he or she is working on(Eby & Freeman, 1999; Beecham et al., 2007). Job characteristic model seems to be frequently used in the empirical studies of existing software for it is ideal to explain the relationship between motivation and work performance through one’s job characteristics(Eby & Freeman, 1999; Hackman & Oldman, 1975; 1976; 1980; Fried & Ferris, 1987).

Therefore, it is necessary to inspect the job characteristic model to understand the motivation and the performance.

In Hackman & Oldham’s job characteristic model, five job characteristics can positively affect workers’ psychological state, so that they eventually improve motivation, job satisfaction and work performance(Hackman & Oldham, 1975; 1976; 1980).

Job characteristics are such factors that they let a worker acknowledge the experienced meaningfulness of the work, the experienced responsibility for outcomes of the work, and the knowledge of the actual results of the work activities. They are composed of skill variety, task identity, task significance, autonomy and feedback.

To measure how much of the five job characteristics a particular job holds, HackFman & Oldham(1975) created Job Diagnostic Survey(JDS) and defined Motivating Potential Score(MPS), which is used to measure how much job characteristics affect a worker’s motivation.

Kulik & Oldham(1985) said the usage of MPS,
a score that indicates the psychological state a worker possesses against his or her work, had been a general practice. It is because using job characteristic model alone is not enough to evaluate how a worker feels about the job.

MPS allows us to measure the psychological state a worker gets from the job and convert it into a numerical score by which we can evaluate the level of one’s potential motivation. The measurement of psychological states is threefold: the meaning of the work one perceives, the responsibility for the work, and the results of the work. A formula to compute the MPS of a job follows as [Figure 1].

\[
\text{MPS} = \frac{\text{(Skill Variety} + \text{Task Identity} + \text{Task Significance)})}{3} \times \text{Autonomy} \times \text{Feedback}
\]

[Figure 1] Motivating Potential Score

III. Research Model and Hypothesis

3.1 Research Model

The goal of this research is to present the usage level of specifically which Agile practices help enhance a developer’s MPS.

On top of that, we will also examine how the motivation of developers using Agile methodology affects the work performance such as productivity, quality and project duration.

[Figure 2] indicates the research model we set up to verify the effects of Agile usage level toward MPS and the effects of MPS toward work performance.

3.2 Hypothesis of the Research

With the Agile methodology, projects are carried out through many practices. That gives us an idea that Agile practices must have a direct connection with the MPS of developers. Agile practices are the detailed guidelines for developers, by which they are expected to have a positive effect on the MPS. Therefore, we anticipate that applying high usage level will yield more impacts on MPS.

By the purpose and the type of developers, we classified the Agile practices into two categories: the management practices and the development practices. The former are mainly focused on communications needed for managing projects and are composed of the practices that execute development plans. The latter are mainly used to raise the quality by inspecting and controlling the program. As developing jobs generally tend to form a direct relationship with the development practices rather than with management practices, it is anticipated that the usage level of development practices should have more
influences than that of management practices does on the MPS. On the basis of the examinations above, the following two hypotheses have been established.

\[ H_1 \]: The usage level of Agile management practices will have positive influences on the MPS of developers.

\[ H_2 \]: The usage level of Agile development practices will have positive influences on the MPS of developers.

A developer’s motivation directly influences one’s work performance and it is a very important component to improve performance in development (Wilson & Hall, 1998; Wilson et al., 2000). Crockburn(2001) pointed out the individual capability is the core in the success of a project and the developing environment should be able to promote it.

Kerstin & Errikos(2006) said Agile methodology enables the developers to accept active user feedbacks and constantly changing requirements, so that it increases users’ satisfaction. Baddoo et al.(2006) insisted if a developer is highly motivated, it should have positive effects on the outcomes of a project such as the productivity and the system quality.

Besides, many other studies have already proven the positive results after adopting Agile methodology like improvements in productivity, system quality and project schedule management(Maurer & Martel, 2002; Muller & Padberg, 2002; Ilieva et al., 2004; Erdogmus & Williams, 2003; Mann & Maurer, 2005). They are all the grounds to support the idea that Agile methodology boosts a developer’s motivation to improve the overall work performance in development.

On the basis of the facts we have examined so far, we have come to make the third hypothesis at the below:

\[ H_3 \]: The MPS of developers will have positive influences on the work performance.

\[ H_3-1 \]: The MPS of developers will have positive influences on the productivity.

\[ H_3-2 \]: The MPS of developers will have positive influences on the system quality.

\[ H_3-3 \]: The MPS of developers will have positive influences on the project duration.

### 3.3 Explanatory Definition of Research Variables

#### 3.3.1. Job Characteristic

To measure the MPS of developers, we used the job characteristics such as skill variety, task identity, task significance, autonomy and feedback. Using JDS(Job Diagnostic Survey), we defined the items for the job characteristic model. We have then fixed and supplemented those items to make understandable to survey respondents.

#### 3.3.2. MPS(Motivating Potential Score)

Once five items of development job characteristics are measured, MPS can be computed by the following formula:

\[
MPS = \left[ \frac{(skill \ variety + task \ identity + task \ significance)}{3} \times autonomy \times feedback \right]
\]

#### 3.3.3. Agile Practices

We investigated the most commonly used practices among Agile developers in Korea, and we narrowed them down to 14 practices, as noted in [Table 1].

#### 3.3.4. Work Performance
Work performance is the outcome obtained from executing software development projects. The productivity, the system quality and the project duration are the considered factors of this item.

Productivity is defined as the level of ratio between the effort invested to perform tasks and the outputs (e.g. amount of codes, work efficiency). The system quality means the level of overall qualities (e.g. stability, reliability, accuracy, reaction speed) that are generally admitted. To measure the level of delay or reduction of project duration, we set the standard to be the amount of time to deliver products to the customers.

Such performance variables can be generally used in IT projects (Linberg, 1999), and they have already been used as performance variables in existing empirical studies of Agile methodology (Mannaro et al., 2004; Ilieva et al., 2004; Mann & Maurer, 2005).

3.4 Sampling Method

For this research, we selected the sample group of developers from four companies in the IT industry: Yahoo! Korea and NHN from web portal companies, and Samsung SDS and Handy Soft from IT service companies. The survey was carried out through both online and offline side by side. First, we explained the purpose of the survey to the managers in the above organizations and were recommended appropriate developers. We then made contact with the selected respondents to carry out the survey.

Out of 100 copies of questionnaires, we collected 82, and excluded 5 copies that contained incomplete data. A total of 77 questionnaires were utilized for the data analysis in this research.

We applied correlation analysis to investigate the relevancy among research variables while performing reliability analysis and validity analysis as well. The causal relationship among the variables in the research model was verified by means of multiple regression.

IV. Empirical Data Analysis

4.1 Sampling survey

[Table 2] briefly shows technical characteristics of sample respondents. 50 percents of the respondents reported they were using Agile methodology, out of which Scrum was taking up the most weight with 29 percents. 81 percents of all respondents appeared to be male.

In terms of the type of companies, 76 percents were working at web portal enterprises. Also, an age group of 30s was forming the largest group with 71 percents while 20s were 26 percents and 40s were as little as 3 percents. By looking at their IT work experiences, the largest group was in the range of more than 4 years and less than 7 years. 87 percents of all respondents turned out to be in the range of more than 1 year and less than 10 years.

<table>
<thead>
<tr>
<th>Concept &amp; Values</th>
<th>Frequency</th>
<th>Percent (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Methodology</td>
<td>Scrum(Agile)</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>XP (Agile)</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Scrum + XP (Agile)</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Structured/Water fall</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td>Object Oriented</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Component</td>
<td>2</td>
</tr>
<tr>
<td>Gender</td>
<td>Male</td>
<td>62</td>
</tr>
<tr>
<td></td>
<td>Female</td>
<td>15</td>
</tr>
<tr>
<td>Company</td>
<td>Yahoo Korea</td>
<td>46</td>
</tr>
</tbody>
</table>
In order to analyze the relevancy among research variables, we evaluated Pearson’s correlative coefficients. As a result, all significant correlative coefficients between two variables with \( p < 0.01 \) appeared to be in the range between 0.416 and 0.795.

Especially, the relatively high correlative coefficients (\( \geq 0.70 \)) appeared in the correlations between the following variables: on-site customer participation and planning game / sprint meeting, daily scrum meeting and planning game / sprint meeting, automatic unit test and continuous integration, and collective code ownership and MPS.

In addition, the result showed there existed positive correlations between variables. Since the value of every correlative coefficient was less than or equal to 0.80, there should be a low possibility that multicollinearity among

<table>
<thead>
<tr>
<th>Variable</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. On-site customer</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Information radiator</td>
<td>.636**</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Same workplace</td>
<td>.295</td>
<td>.174</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Retrospective</td>
<td>.607**</td>
<td>.428**</td>
<td>.327*</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. Short iterative development cycle</td>
<td>.512**</td>
<td>.458**</td>
<td>.035*</td>
<td>.505**</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6. Daily scrum meeting</td>
<td>.557**</td>
<td>.503**</td>
<td>.457**</td>
<td>.574**</td>
<td>.463**</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7. Planning game/sprint meeting</td>
<td>.724**</td>
<td>.553**</td>
<td>.317**</td>
<td>.685**</td>
<td>.565**</td>
<td>.719**</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8. Automatic acceptance test</td>
<td>.086</td>
<td>-.047</td>
<td>.160</td>
<td>.389*</td>
<td>.193*</td>
<td>.228*</td>
<td>.188</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9. Collective code ownership</td>
<td>.014</td>
<td>.052</td>
<td>.009</td>
<td>.430**</td>
<td>.379*</td>
<td>.169*</td>
<td>.220*</td>
<td>.514**</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10. Pair programming</td>
<td>.039</td>
<td>-.027</td>
<td>.229</td>
<td>.389*</td>
<td>.208*</td>
<td>.263*</td>
<td>.170*</td>
<td>.267*</td>
<td>.537**</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11. Automatic unit-test</td>
<td>.181</td>
<td>.138</td>
<td>.047</td>
<td>.605**</td>
<td>.386*</td>
<td>.342*</td>
<td>.465*</td>
<td>.449**</td>
<td>.661**</td>
<td>.267</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12. Test driven development</td>
<td>.231</td>
<td>.269</td>
<td>.142</td>
<td>.477**</td>
<td>.361*</td>
<td>.486*</td>
<td>.401*</td>
<td>.369*</td>
<td>.312*</td>
<td>.261*</td>
<td>.524</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13. Continuous integration</td>
<td>.200</td>
<td>.140</td>
<td>.046</td>
<td>.596**</td>
<td>.228*</td>
<td>.273*</td>
<td>.459**</td>
<td>.325**</td>
<td>.448**</td>
<td>.333*</td>
<td>.795</td>
<td>.617</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14. Refactoring</td>
<td>.313</td>
<td>.291</td>
<td>.214</td>
<td>.556**</td>
<td>.235*</td>
<td>.458**</td>
<td>.408*</td>
<td>.164*</td>
<td>.252*</td>
<td>.396*</td>
<td>.395</td>
<td>.279</td>
<td>.516**</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[Table 2] Analysis of Respondents
measuring variables may appear.

Note: Significance at *p < .05, **p < .01

<table>
<thead>
<tr>
<th>Variable</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>15. MPS</td>
<td>.102</td>
<td>.217</td>
<td>.214</td>
<td>.522</td>
<td>.392</td>
<td>.399</td>
<td>.327</td>
<td>.553</td>
<td>.722</td>
<td>.586</td>
<td>.551</td>
<td>.400</td>
<td>.464</td>
<td>.442</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16. Productivity</td>
<td>.150</td>
<td>.370</td>
<td>-.032</td>
<td>.517</td>
<td>.333</td>
<td>.429</td>
<td>.368</td>
<td>.323</td>
<td>.258</td>
<td>.023</td>
<td>.530</td>
<td>.522</td>
<td>.456</td>
<td>.497</td>
<td>.689</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[Table 3] Descriptive Statistics and Correlations

4.3 Reliability and Validity of Survey

In this research, we used Principle Component Analysis to grasp how precisely each question could reflect the characteristics to be measured. The merit of Principle Component Analysis (PCA) is by applying a minimal number of variables, it can minimize the loss of information and effectively eliminate those variables which interfere with the validity of measurements. We also used Varimax on purpose to clarify categorization of variables. To extract variables, we followed the process of drawing out the variables whose Eigen values were more than 1.0.

To check if questionnaires have consistency when they are used to evaluate a single variable, we applied internal consistency reliability using Cronbach's α coefficient that is widely used. The result showed that Cronbach's α coefficients were between 0.831 and 0.924, which satisfied the usual acceptance level (>=0.6); the questionnaires in the survey is considered reliable enough.
4.4 Hypothesis Testing

H1. The usage level of Agile management practices will have positive influences on the MPS of developers.

To test H1, we set the usage level of Agile management practices as independent variables and the MPS of developers as dependent variables; then we executed multi regression analysis. The analysis report about the individual usage level of Agile management practices that affected the MPS of developers is summarized in [Table 5].

The regression coefficient of the usage level of Agile management practices explains 44.9% of all changes in MPS and it is statistically significant. Therefore, it is considered the usage level of Agile management practices have positive influences on the MPS of developers. According to the analysis result in [Table 5], it is shown that the factors among the usage levels of management practices that significantly affected the MPS were on-site customer and retrospective.

H2. The usage level of Agile development practices will have positive influences on the MPS of developers.

To test H2, we set usage level of Agile development practices as independent variables and the MPS of developers as dependent variables; then we executed multi regression analysis. The analysis report about the individual usage level of Agile development practices that affected the MPS of developers is summarized in [Table 5].

The regression coefficient of the usage level of Agile development practices explains 67.1% of all changes in MPS and it is statistically significant. Therefore, it is considered the usage level of Agile development practices have positive influences on the MPS of developers. According to the analysis result in [Table 5], it is shown that the factor among the usage levels of Agile development practices that significantly affected the MPS was collective code ownership.

![Table 4](image-url) Results of the Reliability and Validity
Dependent Variable: MPS

[Table 5] Regression Results about Impacts of the usage level of Agile practices on MPS

<table>
<thead>
<tr>
<th>Agile practices</th>
<th>B</th>
<th>S.E</th>
<th>Beta</th>
<th>T</th>
<th>p</th>
<th>R</th>
<th>R²</th>
<th>F(p)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Constant)</td>
<td>18.117</td>
<td>15.938</td>
<td>1.137</td>
<td>.265</td>
<td>.670</td>
<td>.449</td>
<td>3.497</td>
<td>(.321)</td>
</tr>
<tr>
<td>On-site customer</td>
<td>-9.791</td>
<td>3.914</td>
<td>-.556</td>
<td>-2.502</td>
<td>.018</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Information radiator</td>
<td>2.150</td>
<td>3.387</td>
<td>.116</td>
<td>.635</td>
<td>.500</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Same workplace</td>
<td>1.748</td>
<td>3.082</td>
<td>.091</td>
<td>.567</td>
<td>.575</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Retrospective</td>
<td>9.465</td>
<td>3.342</td>
<td>.599</td>
<td>2.832</td>
<td>.008</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Short iterative development cycle</td>
<td>5.284</td>
<td>3.468</td>
<td>.271</td>
<td>1.524</td>
<td>.138</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Daily scrum meeting</td>
<td>2.911</td>
<td>3.330</td>
<td>.188</td>
<td>.874</td>
<td>.389</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Planning game/Sprint meeting</td>
<td>-6.87</td>
<td>4.849</td>
<td>-.036</td>
<td>-.142</td>
<td>.888</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Constant)</td>
<td>28.836</td>
<td>6.267</td>
<td>4.601</td>
<td>.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Automatic acceptance test</td>
<td>2.747</td>
<td>1.640</td>
<td>.213</td>
<td>1.675</td>
<td>.104</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Collective code ownership</td>
<td>6.318</td>
<td>2.589</td>
<td>.438</td>
<td>2.440</td>
<td>.021</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pair programming</td>
<td>2.918</td>
<td>2.124</td>
<td>.194</td>
<td>1.374</td>
<td>.180</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Automatic unit-Test</td>
<td>.292</td>
<td>2.875</td>
<td>.023</td>
<td>.102</td>
<td>.920</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test driven development</td>
<td>1.320</td>
<td>1.870</td>
<td>.097</td>
<td>.705</td>
<td>.486</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Continuous integration</td>
<td>-7.75</td>
<td>2.685</td>
<td>-.060</td>
<td>-.289</td>
<td>.775</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Refactoring</td>
<td>3.595</td>
<td>2.090</td>
<td>.223</td>
<td>1.720</td>
<td>.096</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Dependent Variable: MPS

[Figure 4] indicates that the usage level of Agile management practices tends to be higher than that of development practices. We believe that is because management practices are handier to execute than development practices. In fact, the management-oriented Scrum methodology is more rapidly spreading out than the development-oriented XP methodology.

However, according to the result of regression analysis about H1 and H2, the usage level of Agile development practices appeared to have more positive influences on the MPS of developers than the other. It shows that the MPS is more closely related with the application of Agile practices than the usage level of them.

[Figure 4] A Distribution Chart about the usage level of Agile Practices

Out of many Agile management practices, information radiator had the highest usage level while planning game/sprint meeting and daily scrum meeting had the lowest usage level. For Agile development practices, refactoring had the highest usage level while collective code ownership had the lowest usage level.

Information Radiator is used for managing the
progress in development and information sharing. It applies from a memorandum to automatic tools in a variety of methods.

Planning Game/Sprint Meeting requires a lot of time to set the priorities and determine the work to do. For Daily Scrum Meeting, it is difficult that all team members attend the meeting everyday at the same time.

Such demerits are considered to be the reasons why the usage level of Planning Game/Sprint Meeting and Daily Scrum Meeting are deteriorating.

Refactoring is gradually becoming a common way of software development technique regardless of applying Agile methodology.

On the other hand, software developers tend to evade Collective Code Ownership in that they are usually reluctant to share the codes written by themselves. Besides, the ambiguity in the responsibility for code changes makes it even harder to execute in both psychological and managing perspective.

**H 3**: The MPS of developers will have positive influences on the work performance.

**H 3-1**: The MPS of developers will have positive influences on the productivity.

To test H3-1, we set the MPS of developers who are using Agile methodology as an independent variable and the development productivity as a dependent variable; then we executed multi regression analysis. The analysis report about the MPS that affected the productivity is summarized in [Table 6].

The regression coefficient of the MPS explains 15.2% of all changes in the productivity and it is statistically significant. Therefore, it is considered the MPS of developers who are using Agile methodology have positive influences on the productivity.

**H 3-2**: The MPS of developers will have positive influences on the system quality.

To test H3-2, we set the MPS of developers who are using Agile methodology as an independent variable and the system quality as a dependent variable; then we executed multi regression analysis. The analysis report about the MPS that affected the system quality is summarized in [Table 6].

The regression coefficient of the MPS explains 33.1% of all changes in the system quality and it is statistically significant. Therefore, it is considered the MPS of developers who are using Agile methodology have positive influences on the system quality.

**H 3-3**: The MPS of developers will have positive influences on the project duration.

To test H3-3, we set the MPS of developers who are using Agile methodology as an independent variable and the project duration as a dependent variable; then we executed multi regression analysis. The analysis report about the MPS that affected the project duration is summarized in [Table 6].

The regression coefficient of the MPS explains 5.5% of all changes in the project duration and it is not statistically significant. Therefore, it is considered the MPS of developers who are using Agile methodology do not have positive
influences on the project duration.

5.1 Research Summary

In this study, we empirically analyzed how the usage level of Agile practices can affect Agile developers’ motivation and what impacts the MPS makes toward the work performance.

The results are summarized as below:

The first characteristic is about the relevancy between the usage level of Agile practices and the MPS of developers. It was determined that the usage levels of both Agile development practices and Agile management practices had positive influences on the MPS.

Secondly, we analyzed the usage level of Agile practices in Korea. After investigating the usage level of Agile development, it turned out that the usage level of Agile management was higher than that of Agile development. In connection with the recent rising trend of Scrum methodology, we believe that it is not irrelevant with the fact that the Scrum methodology, which is management-oriented, is easier to adopt than the XP development methodology is.

Lastly, we demonstrated the relevancy between Agile developers’ MPS and their work performance. While the MPS had significant influences on the productivity and the system quality, it did not have significant influences on the project duration.

5.2 Significance of the Research

There are a few implications we can present through the empirical results in this study.

First of all, in order to enhance the MPS, we need to concentrate on which practices to select rather than how well to execute them. Unlike other methodologies, Agile methodology is driven by Agile practices. Especially, retrospective and collective code ownership are the Agile practices that can significantly affect the MPS.

Also, although the usage level of management practices is higher than that of development practices, the latter gives more impact on the MPS of developers in the Agile environment. It is because the usage level of development practices is more directly related with job characteristics.

### [Table 6] Regression Results about Impacts of the MPS on the Work Performance

<table>
<thead>
<tr>
<th>Dependent variable</th>
<th>Independent variable</th>
<th>B</th>
<th>S.E</th>
<th>Beta</th>
<th>T</th>
<th>p</th>
<th>R</th>
<th>R² (Adj. R² )</th>
<th>F(p)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Productivity</td>
<td>(Constant)</td>
<td>2.757</td>
<td>.431</td>
<td>.431</td>
<td>6.402</td>
<td>.000</td>
<td>.390</td>
<td>.152 (.129)</td>
<td>6.463 (.015)</td>
</tr>
<tr>
<td></td>
<td>MPS</td>
<td>.017</td>
<td>.007</td>
<td>.390</td>
<td>2.542</td>
<td>.015</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>System quality</td>
<td>(Constant)</td>
<td>2.143</td>
<td>.339</td>
<td>.390</td>
<td>6.323</td>
<td>.000</td>
<td>.575</td>
<td>.331 (.312)</td>
<td>17.811 (.000)</td>
</tr>
<tr>
<td></td>
<td>MPS</td>
<td>.022</td>
<td>.005</td>
<td>.575</td>
<td>4.220</td>
<td>.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project duration</td>
<td>(Constant)</td>
<td>2.638</td>
<td>.370</td>
<td>.575</td>
<td>7.131</td>
<td>.000</td>
<td>.235</td>
<td>.055 (.029)</td>
<td>2.111 (.155)</td>
</tr>
<tr>
<td></td>
<td>MPS</td>
<td>.008</td>
<td>.006</td>
<td>.255</td>
<td>1.453</td>
<td>.155</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

V. Conclusion
than the other is. Hence, when Agile practices are adopted, organizations must put more efforts and investments to the development practices than to the management practices.

Finally, it was proven when Agile methodology is adopted in development, the MPS has significant influences on the productivity and the system quality, whereas it does not affect the project duration. Those development practices that affect the MPS make it possible for developers to find defects within codes and quickly eliminate them by continuous testing activities. That can eventually lead to improving the system quality and reducing time to eliminate defects, so that they all cause positive influences on the overall productivity.

5.3 Limitation and Plans of the Research

The total number of survey participants used in the empirical analysis was only 77 and the number was not sufficient to evaluate how Agile practices affected MPS and work performance. Especially, only 38 of them were Agile developers, which did not reach the level for us to verify and generalize our research model. In Korea, not many of the enterprises have adopted Agile methodology, and that made it difficult to procure a large sample group for the survey. For future studies, it will be necessary to have an enough number of samples to be able to generalize the results of the survey.

Another limitation we faced was that the measurement of the work performance lacked objectivity and consistency since it was solely based upon subjects’ own responses. It is because every individual could have different perspective or standards about the performance. However, previous studies propose that a developer’s subjective judgment about the work performance is more effective than objective measurement tools or indexes. Therefore, when it comes to measuring work performance, it is more desirable to use measurement tools that are based on objective data along with subjective measurements by individual so that we can gain both consistency and objectivity of the work performance.

One of the foremost characteristics of Agile methodology is to stimulate cooperation of teams and interactions among team members. The point of view in this study is that developers’ MPS and their work performance are all standing on the micro behaviors of the individuals. So, we are planning to add new variables to the research model in future studies: effects of Agile methodology towards the cooperation of teams and the teamwork from the viewpoint of the macro behavior. With this new approach, we will be able to establish more inclusive and meaningful research models for Agile methodology.

Reference


[27] Schwaber, Ken, “Controlled Chaos: Living on
the Edge”, American Programmer, 1996.


