IDENTIFICATION OF SUITABLE WEB APPLICATION DEVELOPMENT METHODS FOR SMALL SOFTWARE FIRMS

Moath Husni¹, Fauziah Baharom², and Faudzhiah Ahmad³
¹Universiti Utara Malaysia, Malaysia, ²fauziah@uum.edu.my, ³fudz@uum.edu.my

ABSTRACT. Many development methods have been proposed for developing web application in small software firms. However, these methods have some limitations. This paper aims to identify the suitable development methods for building high quality web application. In order to achieve this objective, a comparative study was conducted on several current development methods. Comparisons were made according to five criteria that include fitted to 10-50 size, simplicity, flexible to change, customer collaboration and quality assurance used measurement program (QAMP). The findings of this paper will be used as a baseline for building a new development methodology for small software firms that emphasize on monitoring.

Keywords software development methods, small software firms, XP, Scrum

INTRODUCTION

Web-based applications have been known to be of high reliability, high usability, more secured, incorporate advanced technologies, takes a shorter time to market, have a shorter product life cycle and required continuous maintenance (Rodriguez et al., 2002).

Many small software firms are involved with developing Web applications (Richardson & Wangenheim, 2007). A small software firm is any organization or company that has approximately 10 to 50 employees (Fayad et al., 2000; Hofer, 2002; Laporte et al., 2005). The current problems faced by these firms include: i) limited resources for development; ii) limited number of available developers; iii) limited staff skills; vi) lack of well-defined development method; and v) limited adoption of Quality Assurance and measurement practices (Fayad et al., 2000; Dangle et al., 2005; El-Sheikh & Tarawneh, 2007; Altarawneh & Shiekh, 2008; Huang et al., 2008; Tarawneh and Allahawiah, 2009; Pusatli and Misra, 2011).

The development methods for building web applications can be categorized into conventional and non-conventional methods. The two types of methods were studied because they are well-known methods and cover the majority of software development methods. Based on the two types of methods, the study will identify the most suitable development method for small software firms.

The remainder of this paper is organized into several sections. In the next section, steps of conducting the research are presented. The findings and discussion section presents answers to the research objectives, while, the conclusion concludes the overall findings.
METHODOLOGY

This study was conducted in two phases: Identification of the current development methods; and identification of the suitable development methods for small software firms.

Phase One: Identification of the current software development methods

In this phase, resources from journals, books, conferences and internet materials from the year 1970 to 2000 were studied. The aim was to identify the current software development methods. Fourteen methods categorized into two groups’ conventional and non-conventional methods were identified.

The methods were Waterfall, Incremental, V-model, Prototype, Spiral, XP, Scrum, Crystal Family Methodologies (CFM), Agile Modeling (AM), Adaptive Software Development (ASD), Dynamic Systems Development Method (DSDM), Feature-Driven Development (FDD), Lean Software Development (LSD) and Rational Unified Process (RUP).

Conventional methods include methods Waterfall, Incremental and V-model. Non-conventional consists of evolutionary methods (Prototype and Spiral) and agile methods (XP, Scrum, CFM, AM, ASD, DSDM, FDD, LSD and RUP). These methods were found to be the well-known development methods in the software industry.

Phase Two: Identification of the suitable development methods for small software firms

This phase was conducted to determine the development methods that are suitable for small software firms. This was done by comparing all 14 methods attained from Phase One in terms of five criteria: fit to 10-50 size, simplicity, flexible to change, customer collaboration and quality assurance used measurement program (QAMP). These criteria were extracted from past studies that were related to software development in small software firms (Huang et al., 2008; Tarawneh and Allahawiah, 2009; Pusatli and Misra, 2011; Rodriguez et al., 2002). Brief descriptions of these criteria are given in Table 1.

<table>
<thead>
<tr>
<th>Criterion</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fit to 10-50 size,</td>
<td>The number of employees ranges from 10 to 50.</td>
</tr>
<tr>
<td>Simplicity</td>
<td>The development method does not require high experience and skills.</td>
</tr>
<tr>
<td>Flexible to change</td>
<td>The development methods should deal with requirements changes.</td>
</tr>
<tr>
<td>Customer collaboration</td>
<td>The development method should involve the customer within the development process.</td>
</tr>
<tr>
<td>QAMP</td>
<td>The quality of process and product should be ensured using a set of metrics applied by monitoring program.</td>
</tr>
</tbody>
</table>

Each criterion can have values of either “Yes”, “Less” or “No”. Each value has a particular score in which, Yes = 3, Less = 2 and No = 1. The total score of each development method was calculated by adding all scores. The lowest score that can be achieved by a method is 5 and the highest score that can be achieved is 15.

Table 2 shows the values of each criterion and the total score of each method. A criterion value was given to each method based on analysis of past literatures.
FINDINGS AND DISCUSSION

In this section, the suitable methods for small software firms are presented.

Suitable Development Methods For Small Software Firms

As mentioned earlier, in Phase Two, the two categories software development methods conventional and non-conventional were compared using five criteria: fit to 10-50 size, simplicity, flexible to change, customer collaboration, and quality assurance used measurement program (QAMP). Table 2 shows the results.

Table 2. Software Development Methods Comparison

<table>
<thead>
<tr>
<th>Method</th>
<th>Criteria Fit to 10-50 size</th>
<th>Simplicity</th>
<th>Flexible to change</th>
<th>Customer Collaboration</th>
<th>QAMP</th>
<th>Total score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conventional Methods</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Waterfall</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>7</td>
</tr>
<tr>
<td>Incremental</td>
<td>No</td>
<td>No</td>
<td>Less</td>
<td>Less</td>
<td>No</td>
<td>7</td>
</tr>
<tr>
<td>V-model</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>6</td>
</tr>
<tr>
<td>Evolutionary</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prototype</td>
<td>No</td>
<td>No</td>
<td>Less</td>
<td>Less</td>
<td>No</td>
<td>7</td>
</tr>
<tr>
<td>Spiral</td>
<td>No</td>
<td>No</td>
<td>Less</td>
<td>Less</td>
<td>No</td>
<td>7</td>
</tr>
<tr>
<td>Non-Convention Methods</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Agile Development Methods</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XP</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>13</td>
</tr>
<tr>
<td>Scrum</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>13</td>
</tr>
<tr>
<td>CFM</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>11</td>
</tr>
<tr>
<td>AM</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>9</td>
</tr>
<tr>
<td>ASD</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>9</td>
</tr>
<tr>
<td>DSDM</td>
<td>No</td>
<td>Less</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>10</td>
</tr>
<tr>
<td>FDD</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>9</td>
</tr>
<tr>
<td>RUP</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>9</td>
</tr>
<tr>
<td>LSD</td>
<td>No</td>
<td>Less</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>10</td>
</tr>
</tbody>
</table>

Sources:
1-Conventional methods sources: (Naqvi, 2007), (Manassar & Govardhan, 2010), (Koblenz, 2003), (Awad, 2005), (Imreh & Rasinghani, 2011), (Okoli and Carillo, 2012).
2-Evolutionary methods sources: (Alite & Spasibenko, 2008), (Koblenz, 2003), (Manassar & Govardhan, 2010).
3-Agile methods sources: (Lindstrom & Jeffries, 2004), (Stojanovic et al., 2003), (Väänänen 2008), (Beck, 1999), (Abrahamsson et al., 2002) (Schwaber & Beedle, 2001).

Based on Table 2, the first method i.e Waterfall received a total score of 7. This is calculated by substituting the value Yes with “3”, Less with “2” and No with “1” and adding all the scores together. Therefore,

\[ \text{No + Yes + No + No + No} \Rightarrow 1 + 3 + 1 + 1 + 1 = 7 \]

The table shows that the total scores for all methods range from 6 to 13. The two lowest scores are found to be 6 and 7, deriving from the conventional type (waterfall, incremental and V-model) and the non-conventional, specifically evolutionary methods (spiral and prototype).
The highest total score is 13, deriving from the agile methods namely XP and Scrum. Other methods i.e CFM, LSD, and DSDM scored 11, 10, and 10 respectively. The rest of the methods (AM, ASD, RUP and FDD) scored 9.

All agile methods concentrate on customer collaboration and requirement change criteria. However, not all of them adopt a measurement program to ensure the quality of the process and product. In addition, four out of nine agile development methods (AM, ASD, FDD and RUP) are found to be complex development methods. XP, Scrum, CFM, LSD and DSDM are identified as either less complex or simple methods.

Table 2 shows that the most suitable methods that can be used for developing Web application in small software firms are XP and Scrum. These methods satisfy four out of five criteria, while the other development methods satisfy less than four criteria. This finding is similar to Ahmad et al., (2012b).

However, XP and SCRUM are shown to be lacking in applying QAMP (Fernandes & Almeida, 2010; Jyothi and Rao, 2011; Qureshi, 2011). Incorporating QAMP in XP can improve XP management practices and at the same time, monitor the development practices in Scrum. This findings support claims of Fritzsche & Keil (2007), and Qumer & Henderson-Sellers (2008) that affirmed that both XP and Scrum need a qualitative and quantitative metrics to monitor the quality of process and product. In addition, Table 2 shows that XP and Scrum are flexible to requirement changes by using iterative development style. However, both XP and Scrum have problems in terms of tracing and reusing requirements (Fernandes & Almeida, 2010; Fritzsche & Keil, 2007).

The design phase for both methods (XP and Scrum) is simple and this is in line claims from Fritzsche & Keil (2007) and Qumer & Henderson-Sellers (2008).

Based on the above discussion, XP and Scrum though are suitable for small software firms, still have some limitations. Thus, this shows that there is a need of a new development methodology for building web application in small software firms based on XP and Scrum. The limitations in both methods can be improved by incorporating QAMP, establishing requirement repository and merging simple design prototype. The proposed enhancement to cover these limitations is also recommended by Ahmad et al., (2011), Ahmad et al., (2012a) and Ahmad et al., (2012b).

CONCLUSION

This paper aims to identify the most suitable development methods for small software firms. Conventional and non-conventional methods were compared based on five criteria namely: fit to 10-50 size, simplicity, flexible to change, customer collaboration and quality assurance used measurement program (QAMP). Results showed that the most suitable development methods to be used for developing web applications in small software firms are XP and Scrum. However, both methods have limitations which justified the need for a new development methodology based on XP and Scrum.

REFERENCES


