Agile Methodology in Practice

Sharifah Syed-Abdullah  
University of Sheffield,  
Regent court, Portobello Street,  
+44 114 222 1870  
s.abdullah@dcs.shef.ac.uk

Mike Holcombe  
University of Sheffield,  
Regent court, Portobello Street,  
Sheffield, S1 4DP, UK  
+44 114 222 1802  
m.holcombe@dcs.shef.ac.uk

Marian Gheorghe  
University of Sheffield,  
Regent court, Portobello Street,  
Sheffield, S1 4DP, UK  
+44 114 222 1802  
marian@dcs.shef.ac.uk

Abstract

This paper presents an early empirical study on Extreme Programming practices employing a qualitative action research method. The study was conducted on university students doing real commercial development projects to gain an insight into the problems faced by the new developers in adopting these agile practices. The aim of the study was to investigate the effects of the XP practices on the quality of software developed. What emerged from the analysis was the positive relationship between number of XP practices employed and the quality of software delivered as perceived by the clients.

Keywords

Agile methodology, action research, content-analysis, empirical study, Extreme Programming, positivist approach, qualitative,

INTRODUCTION

The ever-changing requirements that are a feature of many software development projects require a new approach in developing software. The emergence of agile methodologies seems to be the answer to this phenomena and acceptance of this approach has grown with each year. Tool support has been developed and introduced to encourage the migration to this methodology. However there is still resistance from several quarters especially development teams to adopt this radical approach in software development. The purpose of this study is to investigate the effects of an agile methodology, specifically the Extreme Programming (XP), practices on the quality of the software developed by the teams. The result from the experiments conducted on the Software Hut projects indicated that the clients’ satisfaction has a positive relationship with the number of the XP practices adopted. The Software Hut project teams were divided into 2 categories: XP teams and traditional methods teams but for the purpose of this study, focus was only made on the XP project teams.

METHODOLOGY

Respondents

The Software Hut class consists of the 2nd year undergraduate students from Computer Science and Engineering degrees, and 3rd year students from Math and Computer degree. The computer science students were required to complete all the subjects in Level 1 and the first semester of Level 2 subject before enrolling in the Software Hut class. The subjects which are related to the Software Hut projects are Introduction to Programming, Requirement Engineering, Object Oriented Programming and System Design and Testing modules in the Level 1 and the related subjects in the Level 2 are Functional Programming, Systems Analysis and Design and Database Technology modules. While attending the Software Hut module, the students were also required to attend the Human-Computer Interfaces and Graphical Interfaces module.

In the Software Hut Spring 2002 class, students were divided into teams consisting of 4 to 5 members. With total population of 96 students, there were 20 teams during this study. Teams 1 through 10 were required to use the XP methodology to develop their software, while teams 11 through 20 were allowed to use any traditional Software methodologies. During the second week, all of the students were given team management course to familiarize themselves with managing group project. Throughout the semester, which consists of 12 weeks, XP students in the class were introduced to Extreme Programming. The first lecture on an overview of XP was given during a 2 hours lecture and the XP students were informed of the XP4Real book [5] available online on the lecturer’s website, documents templates.
available in the Genesys internal website and other references about XP practices on various XP websites. The XP teams were also introduced to the practice of pair programming during a lab session at the beginning of the semester. While the XP teams were introduced to the Extreme Programming methodology, the other students were introduced to other techniques used in a traditional methodology. The lectures were conducted as parallel sessions. Lectures on Testing were given to both teams. The unit and functional testing lecture was delivered as part of the overall testing. At the beginning of the semester, each team was required to select their client. Each client was from a local business or organization. For this session, 4 clients selected by the lecturers were:

- Small Firms Enterprise Development Initiative (SFEDI)
- School of Dentistry
- University of Industry
- National Cancer Screening Service

During week 1, each client was required to give a brief introduction of their company and its requirement, to the students. After the briefing, the students were given a week to compare skills among team members with the skills needed to build a solution for each client. After the students made their evaluations based on their strengths and weakness, their choices were submitted to the lecturers. Final decisions regarding clients-students teaming were made by the lecturers to ensure equal distribution. Distribution of clients and teams is shown on

<table>
<thead>
<tr>
<th>Companies</th>
<th>XP teams</th>
<th>Traditional teams</th>
</tr>
</thead>
<tbody>
<tr>
<td>SFEDI</td>
<td>5, 7, 8</td>
<td>18, 20</td>
</tr>
<tr>
<td>School of Dentistry</td>
<td>2, 6</td>
<td>12, 14, 17</td>
</tr>
<tr>
<td>University of Industry</td>
<td>1, 9</td>
<td>11, 13, 19</td>
</tr>
<tr>
<td>National Cancer Screening Service</td>
<td>3, 4, 10</td>
<td>15, 16</td>
</tr>
</tbody>
</table>

Table 1: Distribution of clients and teams in Software Hut 2002.

Table 1.

Every week, each team was required to deliver minutes of each team’s meetings and the timesheet. The timesheet is the document, which was used to collect data for the other study. After week 4, each team was required to submit the Requirement document and the guidelines for this document can be accessed through the lecturer’s website [5]. At the end of the semester, the teams were required to submit working software, software documents and a commentary document. The commentary document is the document where the team members described their experiences developing software with real client using the XP methodology.

**Data Collection**

For this study, 2 methods of data collection were employed to add rigour, breadth and depth to the study.

**Focus group Interview**

The focus group approach was employed in this study because team members develop the software and the existence of team interactions helps to release inhibitions among team members, activate forgotten detail of experiences and also to generate better data through wide ranges of responses. Focus group interviews were also chosen because it is the most appropriate method for studying attitudes and experiences; exploring exactly how opinions were constructed [6] understanding behaviours, values and feelings, [7]. The students were encouraged to choose their own software development team members. The formation of these groups in the Software Hut module was a combination of ‘natural’ groups and ‘constructed’ groups. Natural groups consist of members who have been attending similar modules before attending the Software Hut module or members living in the same residential hall. The ‘constructed’ focus groups were teams, which have fewer team members and required the lecturers’ assistance to form a complete team. Through developing software together, communication and interaction between team members were promoted and improved at an early stage. This aspect of team formation is important to observe, for the study of cohesion and communication between team members. The initial interviews were conducted to get a glimpse of the students’ understanding of the new methodology. The difficulties encountered during the early interview sessions resulted in the development of XP Activities/Product Table 2 [1, 2, 5]. The table was used to explain further about XP practices and the feedback from these interviews were analysed to improve it. The cyclic learning process [4] helped the researcher and the students to understand the
XP practices as it is practiced in the software observatory environment.

The second interviews were conducted after the analysis of the requirements documents were made. For this interview, project plan guideline (Figure 1) and an improved XP Activities table (Table 2) were used as a guide to assist the students in identifying the relevant activities adopted by their team members. In comparison, the time taken to complete the second interviews were shorter and the discussions were more precise. These interview sessions enable the researcher to observe group interaction among the team members.

**Project Assessment Documents**

The assessment for each project was recorded in the project assessment documents. The clients were required to follow a structured marking scheme when awarding the marks to the teams. The clients were briefed before the final presentation during week 12. The assessment documents were filled after all teams have delivered their software.

### Analysis

**Content Analysis**

The interview transcripts and notes taken during the interview sessions provided a complete record of the discussion made with various teams. In addition to the transcripts and notes, project documents were also analysed using the content-analysis method. Traditional content analysis was chosen because it is easy to conduct during the preliminary study. The analysis begins with a comparison of the words used in the interview and the documentary sentences found in the various project documents. The result was recorded in the Individual team checklist and tabulated in Across-group checklist.
RESULTS
This section identifies some of the result of the study conducted on the Software Hut projects.

XP Practices Pattern
The interviews conducted yielded certain patterns among the teams in adopting the 12 practices. The result was verified through the analysis of the documents produced at the end of the semester. Analysis of these documents explained that team members were willing to adopt and adapt practices that were most understood. Of the 12 practices, attempts have been made by every team to adopt:

Planning games
Every team plans their project but writing user stories is another matter. The requirement to include story cards in their documentation forced several teams to attempt writing the stories. In all the documents analysed, none of the clients write a story card. The minutes for meetings among team members for 2 teams indicate that the story cards were developed after the requirements have been captured. This is because the story card idea was taught in the course after the main requirements were captured.

Pair programming
This practice was introduced to the students during a lab session. The knowledge and experience gained during the early lab session help the students in deciding whether to continue with this practice. Feedback obtained during the early interviews gave a negative impression on the acceptance of this practice. The students gave various reasons for not practicing pair programming during the initial interviews but admitted the various advantages associated with the practices in the second interview sessions.

XP methodology commands rigorous testing being done on the Software. Testing in this methodology constitutes Test-first coding, Unit testing and Functional testing. Testing demands teamwork to ensure that testing is done exhaustively. In all the teams (except team 4), the members acknowledge that they adopted pair programming readily during the testing sessions, which is nearly all the time in XP. From the second interview conducted, it was discovered that attempts by students to pair in teams of 3 is a waste of time. Some of the teams with 5 members initially divided themselves in the ratio of 3:2. Discovering that the third member does not contribute productively, the teams usually rearranges the members into 2:2:1 ratio.

Testing
Testing was highlighted as the most important part of the XP methodology. Lectures delivered and explained in detail the procedure and elements needed during testing. Several teams produced detailed descriptions of the tests conducted throughout the projects. Several reports highlighted the importance of this practice and its impact on their project in terms of program integrity.

Collective ownership
Attempts to practice collective ownership were made by several teams. These teams swapped partners and in addition; a team also exchanged modules among the team members. One team did not change the members in each pair but exchanged modules to experience collective ownership among the members. The commentary document analysed indicates the team’s belief that by having the other partners testing their modules will enable the whole team member to understand the overall systems. Among these teams the winning team adopted the 2 activities associated with collective ownership practices.

Frequent releases/ demonstration
Due to the time frame and scale of the Software Hut projects, it is more beneficial to conduct frequent demonstration of the software being developed at different stages. The students were advised to visit the client and show the user interfaces developed during the various stages. The documents analysed revealed that 4 teams attempted to practice frequent presentation. The analysis identified that the 3 teams, which were awarded the highest marks by the clients, constantly embraced this practice. Frequent release or in these projects, frequent presentations gave the teams the benefit of improving communication, understanding and building rapport with the clients.

Coding Standard
There were attempts by several teams to follow certain coding standard. 2 teams referred to the existing coding standard available in the website, while the other 2 teams referred to the Genesys Coding standard available in the
Agile Methodology in Practices

company’s intranet. Genesys is the university Software house. In addition to the Genesys coding standard, the fifth team also identify and formulate its own team standard. In the documents analysed, the only team which did not follow any coding standard, encountered incomplete and incompatibility problem during the development of the Software.

It is difficult to report exactly about the other practices because the absence of guidelines for the commentary report. There is a possibility that other practices were unconsciously practiced because none of the XP practices were new. Most of these practices have been proven over decade (for implementation strategy) or centuries (for the management strategy) [1]. For this reason the Table 1 developed is not conclusive but needs further modification for future study.

The implications of XP practices on Clients’ marks

The radar graph (Figure 2 to Figure 4) illustrated 2 important points in the marking scheme of the Software Hut projects. The graph for overall marking (Figure 2) did not reflect any specific pattern between marks and the number of practices adopted by the teams. The pattern emerged when the marks were divided into clients’ mark and lecturers’ marks (Figure 3 and Figure 4).

It is too early to conclude this, but the pattern in the lecturers’ and manager’s marking scheme revealed that by knowing the students, there is a tendency to look beyond the finished product to measure the students’ capabilities. The radar graph in Figure 3 illustrated there is little effect of XP practices on the marking scheme by the lecturers and managers.

The graph in Figure 4 illustrated a pattern between number of XP practices adopted and the marks given by the clients. The marks obtained were higher for teams, which adopted the most practices (except team 3).

The overall pattern for clients’ marks was analysed further through individual client’s marks. This is important because individual clients set certain standards for their Software product and comparison across clients will not revealed the exact comparison. Analysis on individual client’s marks revealed that the marks awarded by the clients were parallel to the number of practices adopted by the teams (Figure 5 to Figure 8). The line graph reflects a constant relationship pattern between practices and marks awarded.
Agile Methodology in Practices

CONCLUSION
This paper presented an experiment conducted on students embarking on their first full scale software development with real clients. The study follows through the various stages in the software development life cycle, focusing on the adoption of XP practices to facilitate the development process. Adopting an action research approach allows the researcher to intervene because intervention [3] is not considered violation of validity but rather adds to the flexibility of the strategy. The early responses from the students allowed the researcher to advise the teams in using XP practices as close as possible without compromising their decisions. Through collaborative communication processes adopted during the interview sessions, the researcher is able to have an insight into the problems faced in adopting the agile practices. The positivist approach in analysing and reporting the results of this study depict that the more practices were adopted the higher is the quality of the software developed from the clients’ perspective. It is hoped that this research will be able to contribute to the existing efforts of current researchers in promoting effective practices and thus producing productive developers.

REFERENCES


