Sindh Univ. Res. Jour. (Sci. Ser.) Vol.47 (1):09.14 (2015)

SINDH UNIVERSITY RESEARCH JOURNAL(SCIENCE SERIES)

Supporting Agile Methodologies in Practice: A Study of Pakistani Software Industry

T. AZIZ, S. SAEED[++], I. ALSMADI*

Department of Computer Science, Bahria University Islamabad, Pakistan

Received 4th July 2014 and Revised 13th December 2014

Abstract: Agile practice adoption is a complex and costly process for small or medium sized enterprises. Since most of the Pakistani software industry is mainly based on such enterprises, hence it is important to understand the correlation between the two. This study provides a comprehensive understanding of the existing project management processes and their short comings. Based on the study we have developed a web based project management tool to overcome some of the shortcomings which are obstructing the use of agile practices in our local software industry.

Keywords: Software Industry, Agile Practices, Empirical Study, Human Centered Computing, User Centered Design

1. INTRODUCTION

Widespread integration of software systems in everyday life has amplified its scope and applications; however software development is an intricate and human intensive activity. Different models have emerged in order to structure the software development process (cf. Boehm, 1988;Ghezzi *et al.,* 2002;Humphrey, 1989;Pressman, 2005), yet most of these development models proved insufficient for diverse projects. To improve the project success rate, agile methodologies emerged which ensured iterative development and limited focus on documentation, user involvement and active team communication (cf. Bayer and Highsmith, 1994; Highsmith and Cockburn, 2001; De Lucia and Qusef, 2010; Grau, 2012; Saeed *et al.,* 2012; Highsmith, 2013). Agile methodology is an umbrella term used for many approaches such as: Adaptive Software Development,Scrum, Extreme programming, Dynamic System Driven Development, Feature Driven Development, Lean Development asf.

There has been some related work where different researchers have highlighted on different agile approaches (cf. Stapleton, 1997; Palmer and Felsing, 2001; Beck andAndres, 2004;Philip *et al.,* 2004;Poppendieck, 2007; Cristal *et al.,* 2008). However adopting agile methodologies is a complex process and requires sufficient resources and time but there is not much scientific literature focusing on practical issues while adopting agile methodologies. In order to implement agile practices many software applications such as JIRA, Basecamp, Redmine etc. Although these tools have benefited some practitioners but there are issues like cost, feature support, usabilityand compatibility issues. Sarkan and Bakar analyzed using JIRA and Redmine tools in requirement engineering processes of agile projects (Sarkan *et al.,* 201; Naseer *et al.,* 2013).

In this study we have looked at Pakistani software industry to understand the problems in implementation of agile practices and to highlight associated problems.Pakistani software industry is in its nascent stages and most of the organizations are small and medium scale enterprises. We were mainly interested to understand problems faced by such organizations in adopting agileapproaches. Knowing the problems can help in creating prototypes that can improve their work practices.

The rest of the paper is organized as follows: section 2 discusses the methodology used to answer our research question followed by empirical data in section 3. Section 4 discusses the details of our prototype development phase and is followed by evaluation of prototype in section 5 and conclusion in section 6.

2 MATERIALS AND METHODS

We carried out our empirical study in two stages to get answers for proposed questions. In order to understand the abstract problems, firstly we carried out a pre study gathered quantitative data using questionnaires. Later on, we gathered qualitative data by semi structured interviews. For the purpose of questionnaires, we utilized a free online survey application Murvey (Murvey, 2014). The questionnaire was composed of 33 questions. For response, we approached experienced personnel from local companies via the web link. 97 people responded to our survey. The second phase of our empirical study was carried out by interviewing some respondents. This was done to validate previous information and to acquire further details. Based on their easy access, we selected 3 participants for interviews of varying duration. The prime focus of the interviews was to understand the organizational culture of the local industry and the software

[++]Corresponding author E-mail:alsmadi@gmail.com

* College of Engineering, Prince Sultan University, Saudi Arabia tariqaziz.chk@gmail.com

engineering techniques used in them. This helped us to understand the effectiveness of such tools in these companies. One of the aims of this study was to assess the drawbacks and missing features of the existing tools. Data analysis was carried out by using grounded theory and by transcribing the interviews. Patterns were identified from data instead of developing any hypothesis.

After careful analysis of the two studies we came across some requirements. Based on these requirements we have designed a web based prototype for supporting agile project management practices. Yii framework was used for designing the prototype (Yii, 2014). For the verification of the applicability of our proposed solution, two companies from local industry were selected. Their feedback was collected via the same web link after two weeks of usage.

3 RESULTS

In this section we describe the empirical data about our study.

3.1 Pre Study

While developing any tool it is necessary to know the respective expertise of the users. During the survey we have mostly targeted the developers because they play a key role in identifying requirements. Hence it was important to get a developer's feedback while determining tool requirements. Out of 97 respondents 51 % were developers, 20% were managers, 11.65% were CEOs, 10.68% were UCD experts while 5.8% were associated with testing and quality assurance, as shown in Fig. 1: Respondent's position in company. A number of applications are used in the industry to test the performance of the software being developed. These include automated functionality, performance and load testing facilities. In our survey we found that 62.9% companies don't use any particular tool for software testing. In order to clarify requirements 31.9% companies use, use cases, 29.9% companies fix usability requirements with implementation, 28.8% use prototyping while 20.6% depend on contextual task analysis. We found that 57.7% companies use Bugzilla, 28.8% companies 'use Redmine while 13.4% companies are using JIRA, as shown in **(Fig.1 and 2).**

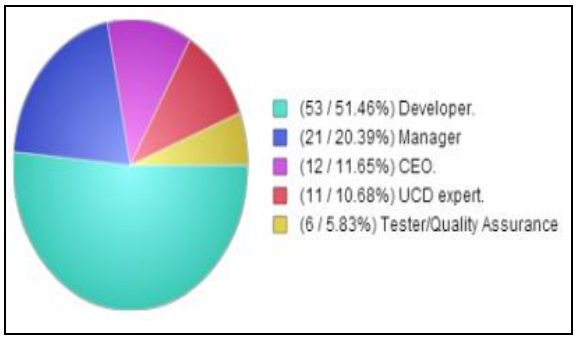


Fig. 1: Respondent's position in company

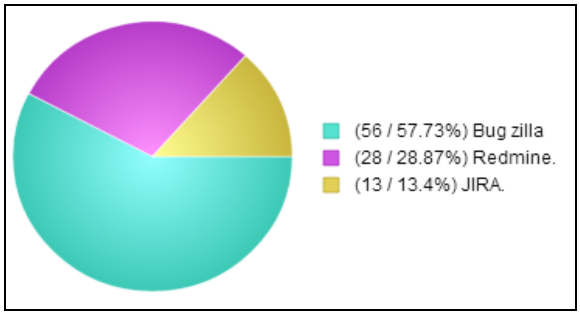


Fig.2:U**sageof Tools**

Most of these tools are not free. Many of the tools which provide some great functionality require license to be purchased. On the other hand free tools are available also but they provide only the basic features. The results of our survey convey that 66% of the participations consider the price nominal while 34% of the users responded that it is expensive. This shows that users are willing to pay for good tools rather than having free tools with limited features. Some of the features of these tools are very helpful for the team members and the customers as well. These include: Planning board, task board, working tasks view, to do, in progress and done tasks view, backlog view, workflow view, drag and drop facility, burn down charts, release board, etc. Determining the most usable features of the tools is very important for development and design phases. In our survey we found that 56% use the tool for workflow view, 54% for task board, 46% use tools for the release board view, 45% for burn down charts facility, 44% for the planning board, 40% for drag and drop facility, 37% for working task view, 35% sprint view and 31% use it for backlog viewing.

Feature driven facility is always a good facility. Mostly the requirements are provided in the form of an epic and then that epic is then used to drive features of the product being developed. Hence requirement gathering and splitting is a vital phase. Requirements are then treated as individual tasks and afterwards the tool processes them. Our survey shows that 52.8% industries already follow this same trend. We found that 60.8% respondents agree that their existing tool is not very effective in getting features while 39.1% says that it is effective for them. In order to assess whether current tools are using elaborate requirements in the form of user stories or not, we included one question related to this issue. According to our survey 86.6% companies agreed that their tools are using elaborate user stories while only 13.4% said that they are not using this feature currently. In order to determine the effectiveness of the tools in describing user stories we included one question related to this in the survey. The response showed that 60.8% respondents agree that their tool is effective in this regard while 39.1% were not satisfied with the performance of their tool. Another important facility is the user story card view. It contains all the possible

options related to the user story at any time. It contains features such as: Type, priority, estimate time; time spent and change history etc. It is the most usable area of the tool and the effectiveness of the tool greatly depends on it. In order to assess the availability of this feature in existing tools our survey showed that 58.7% companies said that their tool has this facility while 41.24% denied its presence.

Task break down for long stories is very important. Multiple smaller tasks make it easier to process and predict estimates. According to results of our survey 61% participants agreed in favor of this feature and 39% were against it. Keeping track of the changes and updates related to the tasks is also very important. According to our survey, 92% respondents said that the changes of the user story description are kept while 8% of the participants said that the history is not kept when the contents are updated. Simply logging changes is not enough as knowing the exact details of the updates is also at times necessary. History logging feature should therefore be detailed.50% of respondents did not agree to this point while 49.8% respondents agree that knowing these details are important. Integration with email system for quick feedback or notification is another facility that can be incorporated in such tool. When asked about this facility 98% of the participants agreed while only 2% were against it.

3.2 Quantitative Interviews

In our pre study we found that there were need of improvements in supporting the complex agile processes. In order to extract detailed requirements of a proposed prototype we carried out semi structured interviews and clustered core requirements based on their relevance. The prime focus of the interviews was to understand the organizational culture of the local industry and the software engineering techniques used in practice.

3.2.1 Adopting a project management tool

In our interviews we found that availability and efficiency of project management tools were big problems. Some tools maybe technically not good but the organizations may not have money so they have no other choice.

3.2.2 Quality Assurance

It was also observed that integrating user stories and testing processes in an application software can improve the working of software teams.

3.2.3 Team Performance Board

Although charts and task sheets are good ways to inspect the project's progress, however to gather the information regarding the daily working of all the members at one single board is an appealing idea. This way one can monitor the performance on daily basis. According to some people, performance evaluation at the team level is important in order to change the organizational culture of helping each other, and not ending up with a group of selfish people who just happen to work for the same project.

3.2.4 Usability Issues

Improved usability can help in better usage of software applications, so this is true for project management applications as well. During our empirical work it emerged that a component which can help in user interface testing can enhance the usability testing process.

3.2.5 Social Networking Integration

Some of the respondents described the need of integration of social networking with project management application. In this way the registration overhead can be reduced, so development team and customers can both use this tool efficiently.

4 PROTOTYPE DEVELOPMENT

In this section we will detail the implementation of design requirements acquired during our empirical study. Keeping those requirements in view we have outlined the basic features of our prototype.

4.1 Electronic Task Board

As observed in data analysis that task boards are vital as they summarize the progress of a project. However, very few tools provide a task board facility. We believe that we can enhance the capabilities of a task board by customizing the status. Increasing the status report and updating the issues on it is another good approach. This enables the team members to utilize a single board to manage their tasks.

4.1 Agile Planning Board

Another important requirement is an agile planning board. Backlog and sprint management is an important feature of an agile planning board. An agile board should facilitate team members to move tasks, to change their priorities and to create sprints. The sprint starting mechanism should be invoked when the sprints are arranged and tasks are included. Team members should be able to view and update the properties of the tasks related to their project.

4.2 Events Manager

Calendar and events manager are two important features. Google event calendar is currently being used by many companies for their event management. It is mainly being used to schedule meetings and to create events. Other tools also provide event management but most of them are read only where users can only view the assigned tasks. Our prototype provides the similar event calendar. It updates the respective team member at the completion of a milestone.

4.3 User Performance Board

In most companies, the managers or the team leaders want to see the progress of their individual team members. For this purpose a performance board is required where various performance measures and the total time spent by individual team members on tasks should be displayed. Our prototype is different from a performance chart as it has various filters for measuring the performance of team members.

4.4 Automatic Issue Creation and Customer Feedback

Customer feedback is vital for the success of any tool. Usually feedback is provided via emails. However it is important to directly include in the task/bug list, the email having feedback. The project administrator should be able provide the receiving end's contact information on demand. The project backlog list of the tool should include related messages via filters. A notification should be sent to the manager/administrator about a new task/bug/issue whenever it arrives.

4.5 Social Network Integration

These days social networking has become a popular communication platform. Our prototype's sign up is integrated with one of the popular social networking sites. This makes it easy for users to sign up using their social networking accounts. The tool then sends related updates to the user's time.

4.6 Task History Log

Keeping track of the changes and modifications in the project is important. Some tools provide this facility while others do not. Tracking history is a main feature of our tool. The main advantage of using our tool is that it will have customizable logging fields.

The structure of the basic entities of our system is explained in **(Fig. 3)**. Our system comprises of 26 entities. The three main entities are: Issue, Users and Projects. Rest of the entities is related to these three entities. The ERD diagram is designed in such a way, that all the artifices are stored in the table, such as: Issue priority, issue status, issue fields, permissions, and groups. **(Fig.3,4)** highlights the agile planning board in our prototype.

5 EVALUATION IN PRACTICE

In order to assess the performance of our prototype in industry, we asked users to try it for one week and then give their feedback using web based feedback form. One respondent commented as follows:

"..the system is quick despite the number of tasks/projects that have been added.."

Another participant gives the response in the following words:

".. I really like the dashboard and the menu structure."

Another Participant gave the response in the following words:

"The project management and project related pages are nicely designed and help to easily manage projects related things." This feedback highlights that the software application was able to benefit the practitioners. The evaluation also highlighted some design issues which need to be addressed in next iteration.

6 CONCLUSION

Our pre study helped to understand the underlying problems faced by Pakistani software industry. These problems were further deeply understood using detailed semi structured interviews. The analysis of empirical data was the main source of requirements for our application.

As evident from the previous section, adopting SCRUM using our tool greatly improves the quality of the projects in the local software industry. Frequent changes in requirements are reduced and requirement management becomes easier. Our tool helped developers to efficiently manage time, projects and resources. It greatly reduced the communication gap between the customer, management and the team. This shows that such tools play an effective role in implementing a specific process model in the local industry.

It was also observed that the projects managed by such tools are completed in time and save cost. It also creates a bridge between all stakeholders thus enhancing team work. Improvement in the usability is also observed as a result of easier inspection and feedback facility. As reported earlier issues with project management increases with the size of the team. With the help of our tool large team management becomes easier.

The drawback of avoiding documentation can be overcome by the history log maintained by our tool. It is easy to understand any feature or module by reviewing the comments and log of the issues. The tool reduces workload on project managers. We can safely conclude that our tool helps in implementation of the three empirical pillars of SCRUM i.e. transparency, inspection and adaptation. Furthermore our tool helps the Pakistani software industry in improving the organizational culture and enables them to introduce a better software development process according to their requirements.

The most encouraging aspect is that the companies which evaluated our tool decided to use it for further large and complex projects. They intend using it for future developments as well. Research shows that tools like project management and testing etc. are important for local software industry. We can however achieve further efficiency by integrating our prototype with other tools.

7 ACKNOWLEDGEMENT

The first author would like to thank Bahria University for its support in completing Master thesis, from which this paper has originated.

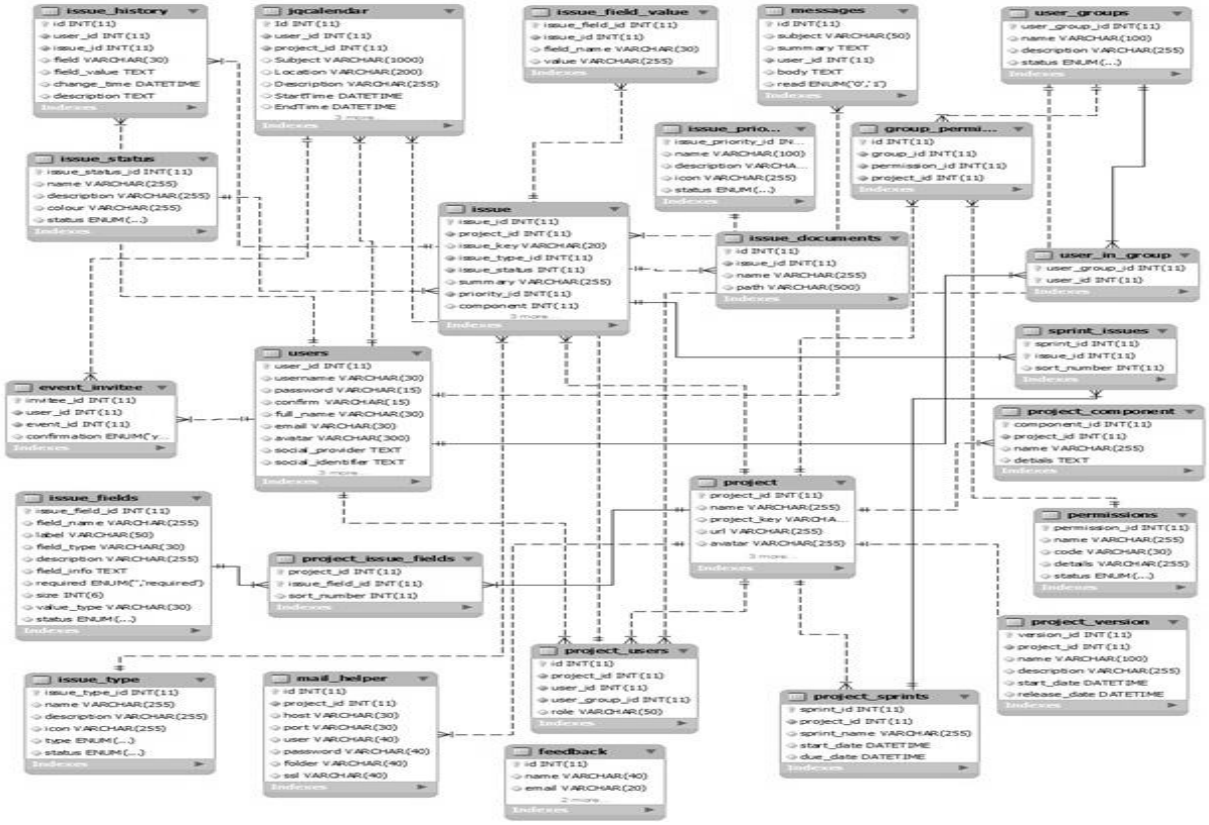


Fig.3: Entity Relationship Diagram

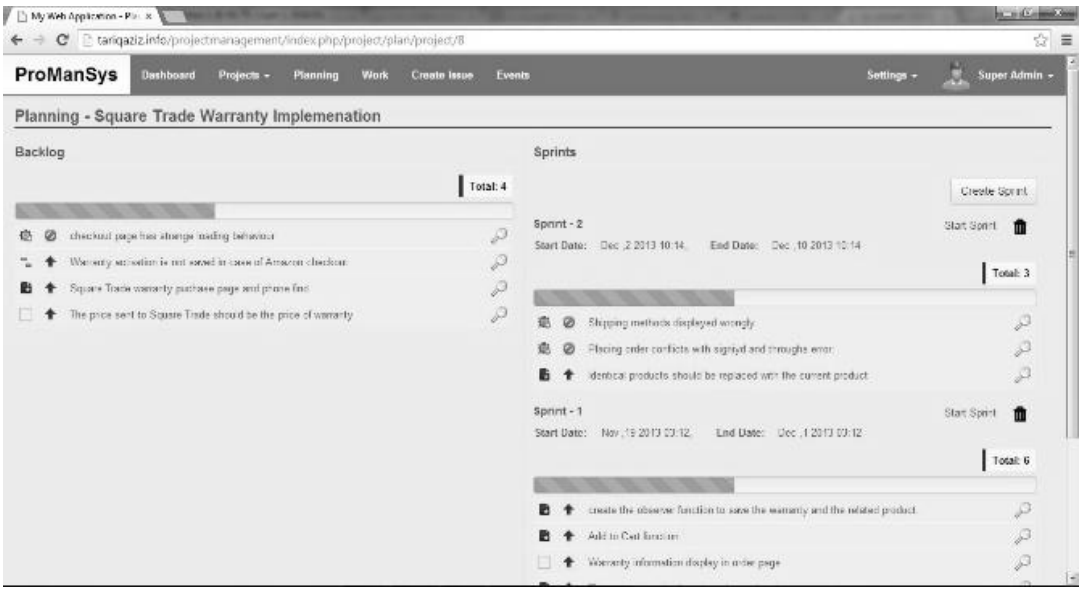


Fig.4: A Snapshot of Agile Planning Board

REFERENCES:

Bayer, S., and J. Highsmith, (1994). Radical software development. American Programmer, 7, 35-35.

Beck, K., and C. Andres, (2004). Extreme programming explained: embrace change. Addison-Wesley Press, USA.

Boehm, B. W. (1988). A spiral model of software development enhancement. Computer, 21(5), 61-72.

Cristal, M., D. Wildt, and R. Prikladnicki, (2008). Usage of Scrum practices within a global company. In Global Software Engineering, ICGSE 2008. IEEE International Conference on 222-226. IEEE.

De Lucia, A., and A. Qusef, (2010). Requirements Engineering in Agile Software Development. Journal of Emerging Technologies in Web Intelligence, 2(3), 212-220.

Ghezzi, C., M. Jazayeri, and D. Mandrioli, (2002). Fundamentals of software engineering. Prentice Hall Publisher USA.

Grau, R. (2012). Requirements engineering in agile software development. InSoftware for People 97-119. Springer Berlin Heidelberg.

Highsmith, J. (2000). Adaptive software development: a collaborative approach to managing complex systems. Addison-Wesley Press, USA.

Highsmith, J. and A. Cockburn, (2001). Agile software development: The business of innovation. Computer, 34(9), 120-127.

Humphrey, W. S. (1989). Managing the Software Process. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA.

Murvey, S. (2014). https://www.murvey.com/ last accessed on 2/6/2014

Naseer, M., H. Rehman, and S. Saeed, (2013). A Survey on Case Tools Adoption for Requirement Management in Pakistani Software Industry. World Applied Sciences Journal, 21(1), 94-98.

Philip, K., P. Eric, and H. C. Betty, (2004). Composing adaptive software. IEEE Software 37(7), 55-64, IEEE Computer Society USA.

Palmer, S. R., and M. Felsing, (2001). A practical guide to feature-driven development. Pearson Education, USA.

Poppendieck, M. (2007). Lean software development. In Companion to the proceedings of the 29th International Conference on Software Engineering (165-166). IEEE Computer Society, USA

Pressman, R. S. (2005). Software Engineering: a practitioner's approach. McGraw Hill Publishers, USA.

Saeed, S., S. Khawaja, and Z. Mahmood, (2012). A Review of Software Quality Methodologies. Advanced Automated Software Testing: Frameworks for Refined Practice, 129Pp.

Sarkan, H., M. Ahmad, and A. A. Bakar, A. A. (2011). Using JIRA and Redmine in requirement development for agile methodology. In Software Engineering (MySEC), 2011 5th Malaysian Conference in 408-413. IEEE Press USA.

Stapleton, J. (1997). DSDM, dynamic systems development method: the method in practice. Cambridge University Press.

Yii, H. (2014) Yii Framework http://www.yiiframework.com/ last accessed on 2/6/2014