A Proposed Hybrid Method for Undergraduate Software Engineering Student’s Project

Abdallah A. Abdallah¹, Abdulwahab L, Bashir S. Galadanci³, Mashal Alqudah⁴, Murtala M⁵

1. School of Computer Science and Information Technology, Linton University College, Malaysia
2. Department of Information Technology Bayero University, Kano, Nigeria
3. Department of Software Engineering Bayero University, Kano, Nigeria
4. Faculty of Information Science and Technology, University Kebangsaan Malaysia
5. Department of Computer Science Northwest University, Kano, Nigeria

Corresponding Authors’ e-mail  abd_wahhb@yahoo.com

ABSTRACT

Agile methods for software engineering has been described as the best for small projects and highly experienced development team. Thus, suggesting that some agile practices requires certain level of expertise and may not be effectively applied by students. Researchers, trainers and institutions are still looking for an appropriate agile method that can be efficiently applied by students and how agility can be achieved in such environment. The objective of this research is to find an appropriate agile method as well as applicable agile concepts, principles and practices that can be suitable for both graduate and undergraduate students’ project. Two software development methods were combined in this study: extreme programming from the agile family and throwaway prototyping from the rapid development methods to come out with an applicable hybrid agile method that can be used by students. The research presented is based on a real time application of certain agile principles and practices in a final software engineering project at Linton University College, Malaysia. As suggested in this research, the hybrid method could guide students in applying agile practices and implementing small and medium scale projects.

KEYWORDS

Agile method, Pair Programming, Extreme programming, Throwaway prototyping

1 INTRODUCTION

Software Development Methods have evolved significantly in the software industry. The emergence of Agile Methods paved way for a faster and more effective ways of developing software systems. Although, the application of agile methods at the university level has been studied to a great extent, for example the research conducted by [1] and [2] studied agile method and practices that could be used by students which focused mainly on Pair Programming and Extreme programming respectively yet students continue to face challenges in adapting agile method as a whole in their projects. The Institute of Electrical and Electronics Engineers (IEEE) suggested agile software development for software engineering programs at the university level and considered it to have a significant importance [1]. Accordingly, the recent call for students to go agile [4] shows that students can take advantage of their projects to apply both technical and methodological aspect of software development.

The difficulty faced by students in applying agile method is the high demand for expertise and team work skills as some researchers [5] and [6] mentioned that agile method are mainly used by highly experienced team of developers. The objective of this research is to find an appropriate agile method as well as applicable agile concepts, principles and practices that can be suitable for both graduate and undergraduate students’ project.

1.1 BACKGROUND

The use of agile software development methods has been staggering over the past few decades [7]. This advancement led to the adoption of agile methods in
software engineering courses in undergraduate classes and subsequently applying it in student’s projects. Some researchers suggested agile development methods such as extreme programming and scrum with little or no modification to their core practices as being adaptable by graduate and undergraduate students [4, 8]. However, it can be argued that agile software development for students requires more than just the application of agile methods [4]. While other researchers such as [9] and [10] describing agile methods as the best for small projects and “highly experienced” development team. This suggested that some agile practices such as pair programming, collective ownership, 40 hours per week (coding) and coding standards etc. requires certain level of expertise and may not be effectively applied by students. Researchers, trainers and institutions are still looking for an appropriate agile method that can be efficiently applied by students and how agility can be achieved in such environment.

2 LITERATURE REVIEW

Numerous researches regarding agile development methods in the context of software engineering education which suggests the need for students to go agile were proposed [4, 8], while some institutions have already adopted agile methods especially in the field of software engineering [9, 10]. Research has shown that there is no single software development method that is the best approach for every project [5]. However, agile methods which include Extreme programming (XP), Scrum, Dynamic System Development, Feature Driven Development etc. have recently gained much popularity in the software industry.

Study has shown that XP is the most widely used method in the agile family [11] and it encompasses all the values from the agile manifesto [12]. Rapid Application Development methods (RAD) which include iterative development, system prototyping and throwaway prototyping are the predecessors of agile methods and the latter is derived from the former [13]. Throwaway prototyping is exceptionally useful among the RAD family and may be used when user requirements are unclear [14] as it encourages the development of a series of design prototypes and other RAD techniques such as Joint Application Development Method Session, Joint Requirement Planning and the use of other computer aided software engineering tools.

A research by [2] explored extreme programming in a university setting. Among the findings include: how XP practices can be applied by students and how it can improve the overall programming experience of the students, while conveying important XP concepts and practices. Although some researchers argued that agile development principles and practices can be difficult to apply by non-agile practitioners [5, 6], this research combined two software development methods with extreme programming from the agile family and throwaway prototyping from the rapid development methods to come out with an applicable agile method that can be used by students. Applicable agile practices which include planning game, pair programming, on-site customer, small releases, refactoring, simple design and testing were selected based on the past studies [2, 8, 9] and were adopted accordingly based on [15] while maintaining agility of the entire process. Agile methods are resistant to frequent changes in specifications during the development process [6] and in most cases turns out to be in a number of iterations before the final product is released [9]. In this study, major emphasis was put on finding out applicable agile practices as well as rapid application development techniques to come out with an appropriate hybrid method for students.

2.1 Extreme Programming Method

Extreme Programming is an agile method which is aimed at speeding development cycle while providing high quality software. Development is done as a series of incremental process or iteration in which the application of XP practices may differ from one project to another [16].Extreme Programming is suitable for small projects with highly motivated, cohesive, stable and experienced team [9]. The XP practices adopted in this research are based on [15] and an overview of the practices is discussed in the subsequent section. The Extreme Programming method is shown in the figure 1.
2.2 Throwaway Prototyping Method

Throwaway prototyping is a Rapid application development method built to develop systems faster than traditional methods such as waterfall [17]. It involves a thorough analysis at the early stage of development which is done concurrently with design and implementation. A small part of the system (prototype) is then developed to explore design alternatives [9]. This could reduce the risk of poorly defined user requirements. Further, once the end-users accept the design prototype, it will be thrown away and design the system again which will then be implemented. Therefore, the prototype will not be used as the final system but rather be used for evaluation by the end-users. However, the final system however, will incorporate all the changes needed in the prototype as shown in the Figure 2.

2.3 Adapting Throwaway Prototyping Method

The concurrent analysis, design and implementation aspect of throwaway prototyping was adapted in the hybrid method. This allows the developers to have thorough understanding of the system and in essence come out with a prototype of the graphical user interface. The final system will then be built upon the prototype which suggests the opposite in the case of throwaway prototyping where the prototype will be discarded and will not be used for the final system. The adapted part of throwaway prototyping is shown in Figure 3.

2.4 An Overview of the Adopted Agile Practices & Iterations Delivery

- **On-Site Customer:** Customers should always be available to provide the system requirements and feedback through all phases of the...
development. They are important part of the development team and would help evaluate the system until it meets their needs.

- **Pair Programming:** Programmers should work in pair and ensure that each line of code is reviewed and understood. The programmers may switch as to who writes the code for a particular function and who reviews it. Other programmers might as well view and inspect the code.

- **Refactoring and Simple Design:** The code should be inspected periodically to ensure coding standards and continuous integration if there are subsystems that were coded separately. The GUI design should be kept simple following what the end-users want and frequent testing should be done to ensure absolute usability.

- **Testing:** The users should be allowed to test each function and provide feedback. User acceptance testing and unit testing may be performed to assess the final product [15].

- **Iterations Delivery:** The system should be delivered in a series of iteration and the end-users should be asked to provide feedback and prioritize new backlog of requirements when completing each and every iteration. However, before delivering first iteration, a clear picture of the system should be shown by conducting JAD session to discuss all the requirements in general.

### 3 METHODOLOGY

The study employ both qualitative and Quantitative methodologies. A qualitative research was conducted to determine the satisfaction of the end-users and to demonstrate the appropriate interfaces and evaluate the final product [4]. The graphical user interfaces were developed based on an empirical case study of Basic English Language Tools for Beginners [9]. While for the quantitative techniques, questionnaire for this study was adapted from the research conducted by [22]. The quantitative technique was conducted by the use questionnaire adopted from [4]. The questionnaire was administered to Basic English language students who are the main users of the system. Statistical Package for Social Science (SPSS) was then used to assess the liability and validity of the questionnaire. The details of the methodology are reported in [22].

### 4 RESULTS AND DISCUSSION

The hybrid method of Extreme programming and throwaway prototyping is relatively an agile method and has most of the principles and practices in the agile manifesto [8]. It focuses on customer satisfaction and system delivery within a short period of time. The system is developed upfront in a series of small iteration and a working system may be provided earlier in the process. There is always on-site customer available to make decisions during development and release of every iteration. Partial pair programming [19] is practiced to ensure each line of code is inspected and reviewed to come out with the system backlog which is used to enhance communication between the developers and customers. The developers also take responsibility of quality and testing of the system as well as refactoring of code to ensure coding standards as determined by the developers [20].

Joint Requirement Planning and Joint Application Development Session were the main Rapid Application Development techniques used in designing and evaluating the prototype. The GUI prototype serves as a basis for deriving the new system. In essence, the resulting GUI prototype will not be thrown way as in the case of throwaway prototyping but will rather be used to code the new system.

Finally, when the customers agree with the GUI prototype, the developers can proceed with the coding of the system and present a working system at the end of each iteration. The hybrid method is shown in the Figure 4.

**Fig. 4. The Hybrid Method of “XP and Throwaway Prototyping**
In the planning phase, a tentative feasibility study of the project was analyzed to understand its technical, economical and organizational feasibility and the project team developed a work plan to set the project on the go. As with extreme programming [21], analysis, design and implementation are done concurrently but the output would not be the system which is contrary to XP. In this methodology, the output of these phases will be the GUI of the system. Comparatively, the output of throwaway prototyping is a design prototype which is not used as the final system but in this methodology, the GUI prototype will be used for the final system. The GUI is therefore developed just to ensure that the developers are doing exactly what the end users are expecting because they will be involved in the process. This is even more important because it is difficult to keep users on-site during the entire development process and users can possibly get bored of waiting for the programmers to code the system and may result to a false feedback or make the end-users uncomfortable. If the end-users are not satisfied with the GUI, the developers will go back to the analysis, design and implementation phases to come out with a new GUI and the process goes on and on until the requirements are fulfilled. The next step is to start coding the functionalities of the system following an essential XP practice which Sommerville [18] called it “Pair Programming”. This means that the developers will work in pair and provide support for each other to come out with the final system. However, if the users are not satisfied with the final system, the developers can go back again to the analysis phase proving that changes of the requirements are fully supported by this methodology. On the basis of stated procedure, extreme programming and Throwaway Prototyping are combined to come out with hybrid agile method which can be used in implementing small and medium scale systems [10].

5 CONCLUSION

This research presents how Hybrid Method of XP and Throwaway Prototyping may guide software engineering students’ project while applying XP practices and Throwaway prototyping techniques. The agile practices include planning game, pair programming, partial on-site Customer, small Releases, refactoring, Simple Design, and Testing while the throwaway prototyping techniques are Joint Requirement Planning and Joint Application Development Session. Findings from this study suggest that the Hybrid Method can guide software engineering students in applying agile practices in their projects. However, there is need for further studies on this method with regards to students’ collaborative skills particularly on pair programming and on-site customer when using the hybrid method.

6 REFERENCES

13. Bayer, S., Highsmith, J. Radical software
development, American Programmer, vol. 7 (1994).


18. Awad, M.A. A comparison between agile and traditional software development methodologies University of Western Australia (2005).


