

A Hybrid Agile model using SCRUM and Feature Driven Development

S. S. Tirumala

Department of Computing
Unitec Institute of Technology
Auckland, New Zealand

Shahid Ali

Department of Computing
Unitec Institute of Technology
Auckland, New Zealand

Anjan Babu G.

Department of computer science
Sri Venkateswara University
Tirupati, India

ABSTRACT

Migration is considered as crucial task especially in case of distributed projects. Banking domain, which involves sensitive business operations, is one such example which involves large volumes of data and transactions. Banking process is as old as history of human transactions, technological migration is necessary to cope up with the latest advancements to provide better service to the customers. For such a task, a single project management method deployment may not be strategically efficient, especially in terms of maintaining schedule and quality. For instance SCRUM and Feature Driven Development (FDD) are two well-known and popular project management methods with their own advantages and shortcomings. SCRUM, being tight with schedules, often quality may be compromised whereas FDD being quality rich, could overrun project time-line.

In this paper for the first time, SCR-FDD, a purposeful amalgamation of SCRUM and FDD methodologies is proposed. The proposed SCR-FDD is evaluated against SCRUM and FDD within a real time project and the results show that proposed SCR-FDD is ten percent more efficient in terms of quality and customer satisfaction compared to its nearest counterpart, SCRUM.

General Terms

Agile project management, Software Engineering

Keywords

SCRUM, Feature Driven Development, Hybrid agile methods

1. INTRODUCTION

Agile methods, with their ability to cope with changing requirements of software development, have received much attention from academicians and practitioners and proved to be effective with volatile functional and non-functional requirements [1][2]. These methodologies are lightweight, in that, they are characterized by a such attributes [3] [4] as adaptive, incremental, cooperative and straightforward. Agile methods, basically an umbrella term, encompass quite a few different approaches such as Adaptive Software Development [5] Agile Modelling [6], Crystal Family[7] [8], Dynamic System Development Method [9], [10], Extreme Programming [11] [12], Feature Driven Development [13] [14], In-

ternet Speed Development [15] [16], Pragmatic Programming [17] and SCRUM [18] [19].

Considering the strengths and weaknesses of various methods, the mutation of the two methods has been a possible solution to overcome weakness of one method by replacing them with strengths of other method. Since this process is domain based, the selection of project management methods is of high importance. But, there are no hard rules to follow, other than successful implementation with minimum implications. In this regard a hybrid approach is proposed using SCRUM and FDD called SCR-FDD giving equal importance to quality and schedule.

The paper is structured as follows. Section 2 introduces SCRUM and FDD in brief. The proposed SCR-FDD is presented in Section 3 followed by its evaluation as Section 4. Finally, conclusion and future work is presented as Section 5.

2. LITERATURE

"SCRUM is a simple process for managing complex projects" [19]. SCRUM is considered to be an empirical methodology for maximizing the Return on Investment (ROI). The SCRUM Master acts as a bridge between technical and non technical people. Sprint planning meeting and daily SCRUM are the two crucial aspects of SCRUM.

Daily SCRUM is an important part of the SCRUM process where in a discussion regarding work processes will be taken up to clarify most of the project aspects instantly. However, they last for less than 30 minutes , but they are significantly important as the daily updates are maintained by SCRUM Master which makes the entire process reviewable. The practices of SCRUM are shown in Table 1. The main advantages of SCRUM is code integration and implementation with timely release. However, the major drawback of SCRUM is setting priority to schedule over quality.

The initial step in FDD is designing an overall project model which none of the other agile methodologies have. A comprehensive list of features to be developed is created based on the design. After this, the entire process becomes iterative. Each feature is assigned to a small team typically with one or two members who owns the responsibility of the feature as a feature owner. Development and Inspections are carried out with respect to the quality. Completed features are released for the deployment. Plan, Development, Inspection and Release are continuous iterative processes throughout the project. List of FDD practices are presented as Table 2.

Table 1. SCRUP practices

Practice	Details
Scheduling	Scheduling is adaptive such that resources can be allocated dynamically at run time
Development	Because of adaptive nature short release deliveries are possible which enables proper user familiarity with the system
Nature	Iterative particularly boxed for small interval
Development	End to end function slices which makes system less complex and function able
Code Integration	A continues process through the project, making testing phase marginally easy
Testing	Ongoing process as you go to avoid backward fall
Programming	Is viewed as design of the system taking care of each stages and division
Implementation	Ready to go as most of the phase include this as a part

The key advantage of FDD is its ability to produce fully functional solutions that requires little or no future attention. In addition to this, FDD also produces workable and tangible independent features for every iteration. FDD operates in feature groups. A feature group is a small team of individuals dedicated to the development, test and release of a feature. This particular working environment unleashes the full potential of the individuals as well as collaborative capability of the teams. However, FDD based projects are unpopular for being out of schedule probably due to undivided attention to quality. As SCRUM is a schedule based approach with strict focus on time line, it would be an obvious choice for FDD to adopt SCRUMs ability of scheduled delivery.

Table 2. FDD Practices

Practice	Details
Designing Model	Development of an overall (high level) objective model for the project
Build Feature List	Creating a comprehensive list with all identified features
Plan (On going)	Creating a development plan and assigning the ownership of each feature
Development (On going)	Building the features
Inspection (On going)	Quality assurance
Release (On going)	Timely releases of workable features

There are a number of attempts to mutate different agile methods within themselves in order to evolve a better methodology on the need of quality design [20] [21]. One such example is a modified approach of XP proposed by Qureshi and Hussain [22] [23]. Similarly, Petersen and Wohlin developed a process model and mapped its properties with incremental and iterative development (IID), SCRUM and XP [24]. Another hybrid model of XP and Throw-away Prototyping was proposed for computer aided design [25]. In 2014, an agile method with combination of SCRUM, XP and The

Dynamic Systems Development Method (DSDM) was proposed to overcome the challenges in software development projects [26], which shows the continued interest in the development of hybrid models.

Most of the hybrid models are SCRUM and XP based and there has been very little research involving FDD. Hybrid models have not always been successful. However, the past failures in implementing hybrid methods should not deter us from using the same [27], which is the motivation for the current theme of this paper.

3. PROPOSED MODEL SCR-FDD

The initial process of SCR-FDD is the identification of features, followed by iterative development of modules consisting of multiple features. Each module (Sprint) is treated as a separate project with a SCRUM Master (Business Analyst) to make sure the scheduled delivery with least discrepancies. Features that developed as a part of module are assessed by leaders of other modules acting as product owners for compatibility issues. With this approach neither of the schedule or quality can be compromised since it is monitored hierarchically. Issues with schedule and / or quality are addressed at feature level. This approach further gives an opportunity to reduce the comparability and deployment issues which are often considered as major obstacles of implementation and release. The next level of process is timely releases of independent features and modules. Project Manager monitors the entire process as an external moderator determined for the successful and efficient project delivery.

This practice of feature based collaborative module approach closely resembles the Service Oriented Architecture (SOA) based development process where individual modules are developed separately, followed by a collaborative process to construct the entire product.

Table 3. Tasks of proposed SCR-FDD

Task	Purpose	Approach
Identification	Identify features	FDD
Build	Iterative development process similar to Nature of SCRUM with assigning small tasks to individuals	FDD and SCRUM
Code Integration	Process over seen by SCRUM Master (Business Analyst) acting as a Product owner	SCRUM
Quality Assurance	Inspections by Business Analyst for quality	FDD
Daily SCRUM	Sprint Meeting after inspection	SCRUM
Backlog Reinforcement	Continues as you go way which makes no backward fall	SCRUM
Release	Timely build and release of workable feature	FDD and SCRUM
Migration	Replacing the existing feature with new features and Inspection by Testing Team and Project Manager	

4. EVALUATION AND DISCUSSION

The proposed SCR-FDD model is evaluated on a product development for a corporate banking company. The corporate banking giant has 37 servers across Europe with 1,800,000 data transactions per day with approximately 370000 customers across the globe.

Table 4. Results of the research

Week Criteria	TEAM SCRUM					TEAM FDD					TEAM SCR-FDD				
	1	2	3	4	T	1	2	3	4	T	1	2	3	4	T
Tasks Completed (out of 20)	6	5	5	4	20	2	3	3	2	10	6	5	5	4	20
Approximate KLOC	3.4	3.2	1.8	1.3	9.7	1.2	1.7	1.1	0.9	4.9	2.9	2.7	2.6	1.2	9.4
No. of Test Cases	35	35	33	25	128	35	35	33	25	128	35	35	33	25	128
Pre-Release Defects	21	15	17	11	64	4	6	9	5	24	16	11	13	6	46
Post-Release Defects	12	9	11	5	37	1	1	3	2	7	4	3	6	1	14
Completion (%)	100	100	100	100	100	34	44	45	50	43.25	100	100	100	100	100
Customer Satisfaction (%)	72	71	71	70	71	42	45	44	41	43	75	77	87	85	81

Table 5. Structure of the Project Team

Role	Size
Project Director	1
Senior Project Manager	1
Team Leaders	4
Business Analysts	4
Developers	27
Testers	8
Total	45

There are eight development centers with two in India and one each in the UK, France, USA, Germany, South Africa and DUBAI. To evaluate the proposed SCR-FDD against SCRUM and FDD, three modules are identified in the existing software. The identified modules are with the same complexity and severity. The existing software was in C++ and VC++ with SQL Server 2005 and the proposed migration is to Microsoft C# and SQL Server 2012. For the current study, the migration of SQL Server is ignored. Three teams of same size are constituted (refer Table 6) for developing one module each using SCRUM, FDD and SCR-FDD. All the teams are equipped with individuals of similar technical and domain expertise evenly distributed.

Table 6. Structure of Teams for evaluation

Role	Size
Team Leader	1
Business Analyst	1
Developers	6
Testers	1
Total	9

Senior members of each team possess considerable experience in both SCRUM and FDD. A workshop is conducted for those with no proper experience in agile development environment. Team leaders are made responsible for the successful delivery of the module and are provided with direct access to the clients. Since the projects are live, a timely reporting on the progress and implementation to the project manager has been mandated. The entire process is scheduled for four weeks and the observations are presented in Table 4. There are 20 tasks in total to be accomplished by each team within specified schedule. The task completion analysis is shown in Fig. 1. Team-SCRUM and Team-SCR-FDD were able to complete all the tasks within the time periods, whereas, Team-FDD was able to complete only 10 tasks out of allocated 20 (50%) and the reason being the nature of FDD to achieve high accuracy with least defects.

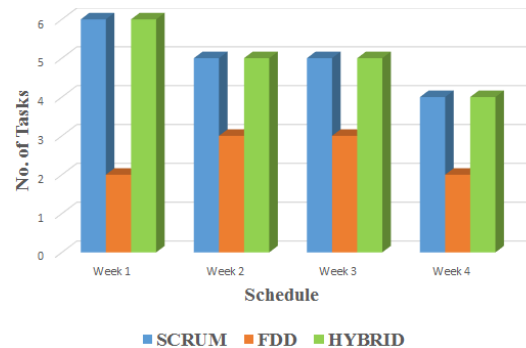


Fig. 1. Task Completion

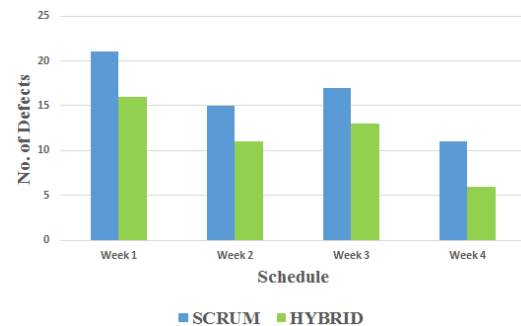


Fig. 2. Pre-release defects

The Pre-release and Post-release defects are represented in Fig. 2 and Fig. 3 respectively. In this comparison of defects, Team-FDD was ignored due to failure of completing all the tasks. With the above results, it is quite evident that Team-SCR-FDD has less number of defects in both the scenarios (Pre and Post). It may be noteworthy to observe that for the ten completed tasks, Team-FDD has only 24 pre-release defects and 7 Post-release defects reflecting how FDD is a quality based approach. The advantage of proposed SCR-FDD approach is clearly demonstrated with the least number of defects among all three influenced by FDD process. The success of any project is measured by customer satisfaction and this project is no different in this aspect. A Customer satisfaction survey was conducted about Schedule, Planning, Intermediate releases, Communication, Code standards and documentation. Usually documentation is ignored in most of the project evaluations. However, it reflects team members' understanding of the project; hence, it is of great importance for these types of evaluations.

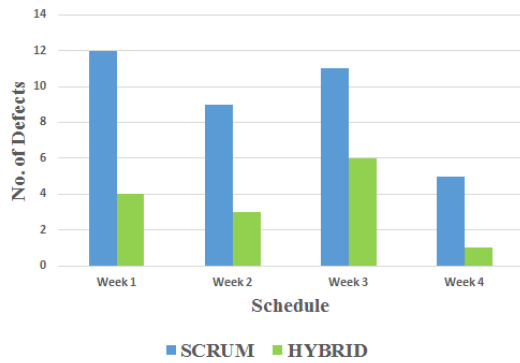


Fig. 3. Post release defects

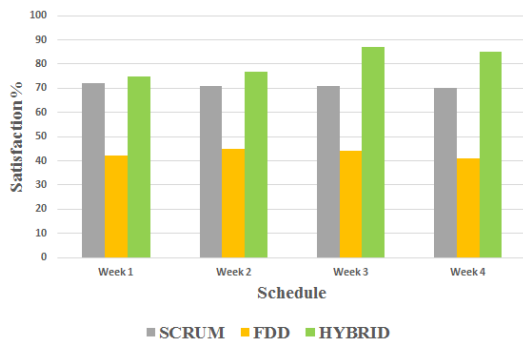


Fig. 4. Customer Satisfaction Report

A cumulative values of the individual statistics are presented in Table 4. From Fig. 4 it can be concluded that the proposed SCR-FDD achieved better customer satisfaction than its counter parts, especially with a margin of 10% in case of SCRUM model. The major difference between SCRUM based approach and proposed SCR-FDD is with defect percentage which has justified the amalgamation of SCRUM and FDD as proposed.

5. CONCLUSION

This paper proposes, for the first time, a new hybrid project management method (SCR-FDD), which is an amalgamation of SCRUM, a schedule oriented approach and FDD, a feature based approach. Since the primary pitfall of FDD is its lack of ability to deliver on time, in this research an attempt to implant SCRUM-like monitoring tasks into FDD is made. This hybrid model has been evaluated in a real time project with three different teams, implementing SCRUM, FDD and SCR-FDD respectively.

The investigative results show that SCR-FDD has a 10% of customer satisfaction over its nearest competitor, i.e. SCRUM, the reason being the less number of defects (both pre-release and post-release). The proposed SCR-FDD was implemented in a controlled environment by treating each module as a project in a small team. However, the SCR-FDD implementation needs to be evaluated on larger scale projects to justify the proposed model and to unleash the complete potential of SCR-FDD.

6. REFERENCES

- [1] F. K. Y. Chan and J. Y. L. Thong, "Acceptance of agile methodologies: A critical review and conceptual framework," *Decis. Support Syst.*, vol. 46, no. 4, pp. 803–814, Mar. 2009. [Online]. Available: <http://dx.doi.org/10.1016/j.dss.2008.11.009>
- [2] C. V. Geambasu, I. Jianu, I. Jianu, and A. Gavrilă, "Influence factors for the choice of a software development methodology," *Journal of Accounting and Management Information Systems*, vol. 10, no. 4, pp. 479–494, 2011.
- [3] A. Cockburn, *Agile Software Development*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2002.
- [4] P. Abrahamsson, J. Warsta, M. T. Siponen, and J. Ronkainen, "New directions on agile methods: A comparative analysis," in *Proceedings of the 25th International Conference on Software Engineering*, ser. ICSE '03. Washington, DC, USA: IEEE Computer Society, 2003, pp. 244–254. [Online]. Available: <http://dl.acm.org/citation.cfm?id=776816.776846>
- [5] J. A. Highsmith, III, *Adaptive Software Development: A Collaborative Approach to Managing Complex Systems*. New York, NY, USA: Dorset House Publishing Co., Inc., 2000.
- [6] S. Ambler, *Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process*. New York, NY, USA: John Wiley & Sons, Inc., 2002.
- [7] A. Cockburn, "Writing effective use cases, the crystal collection for software professionals," 2000.
- [8] —, *Surviving Object-oriented Projects: A Manager's Guide*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1998.
- [9] D. Consortium, *Dynamic System Development Method, Version 3*. Cambridge University Press, 1997.
- [10] J. Stapleton, *DSDM, dynamic systems development method: the method in practice*. Cambridge University Press, 1997.
- [11] K. Beck, "Embracing change with extreme programming," *Computer*, vol. 32, no. 10, pp. 70–77, Oct 1999.
- [12] K. Beck and C. Andres, *Extreme Programming Explained: Embrace Change (2Nd Edition)*. Addison-Wesley Professional, 2004.
- [13] P. Coad, J. d. Luca, and E. Lefebvre, *Java Modeling Color with Uml: Enterprise Components and Process with Cdrom*, 1st ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1999.
- [14] S. R. Palmer and M. Felsing, *A practical guide to feature-driven development*. Pearson Education, 2001.
- [15] M. A. Cusumano and D. B. Yoffie, "Software development on internet time," *Computer*, vol. 32, no. 10, pp. 60–69, 1999.
- [16] R. Baskerville and J. Pries-Heje, "Racing the e-bomb: How the internet is redefining information systems development methodology," in *Realigning research and practice in information systems development*. Springer, 2001, pp. 49–68.
- [17] A. Hunt and D. Thomas, *The pragmatic programmer: from journeyman to master*. Addison-Wesley Professional, 2000.
- [18] K. Schwaber, "Scrum development process," in *Business Object Design and Implementation*. Springer, 1997, pp. 117–134.
- [19] K. Schwaber and M. Beedle, *Agile Software Development with Scrum*, 1st ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.

- [20] J. Highsmith, *Adaptive software development: a collaborative approach to managing complex systems*. Addison-Wesley, 2013.
- [21] A. Cockburn and J. Highsmith, "Agile software development: The people factor," *Computer*, vol. 34, no. 11, pp. 131–133, 2001.
- [22] M. R. J. Qureshi and S. Hussain, "A reusable software component-based development process model," *Advances in engineering software*, vol. 39, no. 2, pp. 88–94, 2008.
- [23] —, "An adaptive software development process model," *Advances in Engineering Software*, vol. 39, no. 8, pp. 654–658, 2008.
- [24] K. Petersen and C. Wohlin, "A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case," *Journal of systems and software*, vol. 82, no. 9, pp. 1479–1490, 2009.
- [25] M. Alqudah and A. Abdallah, "Implementing computer-aided language learning tool using hybrid agile method: A case study," in *Informatics and Creative Multimedia (ICICM), 2013 International Conference on*, Sept 2013, pp. 174–180.
- [26] S. Sultana, Y. Motla, S. Asghar, M. Jamal, and R. Azad, "A hybrid model by integrating agile practices for pakistani software industry," in *Electronics, Communications and Computers (CONIELECOMP), 2014 International Conference on*, Feb 2014, pp. 256–262.
- [27] H. Kniberg and R. Farhang, "Bootstrapping scrum and xp under crisis a story from the trenches," in *Agile, 2008. AGILE '08. Conference*, Aug 2008, pp. 436–444.