

# A Comparative Analysis of DXPRUM and DSDM

Muhammad Fahad<sup>1</sup>, Salman Qadri<sup>1</sup>, Saleem Ullah<sup>2</sup>, Mujtaba Husnain<sup>1</sup>, Rizwan Qaiser<sup>3</sup>, Shehzad Ahmed Qureshi<sup>4</sup>, Waqas Ahmed<sup>5</sup>, Syed Shah Muhammad<sup>4</sup>

<sup>1</sup>Department of Computer Science & IT, The Islamia University of Bahawalpur, Pakistan.

<sup>2</sup>Department of Computer Science & IT, Khwaja Fareed University of Engineering & IT, Abu Dhabi Road, Rahim Yar Khan, Pakistan.

<sup>3</sup>Department of Computer Science, National University of Computer & Emerging Sciences, Lahore, Pakistan.

<sup>4</sup>Department of Computer Science, Virtual University of Pakistan.

<sup>5</sup>Institute of Computing, Bahauddin Zakariya University, Multan, Pakistan.

## Abstract

Agile methodologies are taking over the traditional software development methodologies with the passage of time. These methodologies involve more benefits than traditional software development methodologies and very fewer drawbacks. Due to these benefits, the software development teams are more involved in Agile methodologies. There are a number of Agile methodologies which are famous in software development community. Each of them has its own benefits and drawbacks. Proposing one suitable Agile methodology for a project may benefit the project in one direction but may also cause the project to suffer in some other direction. Due to this fact, the idea of hybrid Agile methodologies evolved. DXPRUM is also one of those hybrid methodologies and is a combination of DSDM, XP, and Scrum. This paper presents a comparative study of DXPRUM and DSDM by applying them to some real time projects and thus comparing their results.

## Key-words:

*Agile, Hybrid Methodologies, XP, RUP, Scrum.*

## 1. Introduction

'Agile' as the name suggests, are fast ways for developing software with minimum budget and without compromising the quality. Creating high-quality software within low budget and with a minimum amount of time is always a crucial task. Agile methodologies are also better over traditional iterative or incremental methodologies when it comes to large and ambiguous requirements which lead to low-quality software. According to Edeki, breaking the large requirements down into more manageable sub-requirements, the Agile process naturally promotes better estimation [1]. As a result of the overall quality of the software increases. This is the reason why more and more software companies are converting themselves from traditional software development methodologies to Agile. According to a survey conducted by the US and Europe in 2005, it is revealed that 14 percent of companies were using agile methods at that

time, and further 49 percent of the companies aware of agile methods were interested in adopting them [2] and these numbers are improving day by day. After having some reading of Agile methodologies, it comes to mind that Agile methods such as XP and Scrum can be viewed as a reaction to traditional and plan based methods, which emphasized an engineering-based and rationalized approach[3]. On the other hand, Agile addresses unpredictability; creeping requirements by focusing much on talented people and their relationship bring to software development [4]. Now with the passage of time, researchers are more focused on combining the advantages of different methodologies in order to get best out of them. This idea is not only limited to combining Agile methodologies, but it also involves traditional software development methodologies like a waterfall or iterative methods. Different researchers proposed hybrid methodologies which are proved to be quite good when it comes to high-quality software. This research work is based on a comparative study of a hybrid Agile methodology DXPRUM with another famous Agile methodology DSDM and to reach a conclusion by comparing their results.

## 2. Materials and Methods

Hybrid Agile methodologies are gaining popularity due to the fact that they involve best rules and practices of the combined methodologies and by leaving their drawbacks. Many researchers in recent past work on different Agile methodologies and proposed hybrid models. Mushtaq and Qureshi (2012) proposed a novel hybrid model by combining two Agile methodologies which are Scrum and XP [5]. Similarly, Bashir and Qureshi (2012) proposed a hybrid software development approach for small to medium scale projects by combining the best practices of RUP, XP, and Scrum [6]. Qureshi (2012) proposed

another enhanced hybrid model and named it eXSCRUM by combining XP and Scrum [7]. Rasool et al, (2013) proposed eXRUP which is a hybrid software development model from small to medium scale projects [8]. Sharma and Wadhwa (2015), proposed eXSRUP which is a hybrid software development model integrating Extreme Programming, Scrum and rational Unified Process. Malhotra and Chug (2013), in their model IXSCRUM, combined XP with Scrum [9]. XSR is a novel hybrid software development model proposed by Ahmad et al, (2014) by integrating XP, Scrum and RUP [10]. All these related work is a good proof of the fact that hybrid Agile models are gaining their roots in the software industry. By keeping all this in mind, Fahad et al, (2014) proposed a hybrid Agile Model DXPRUM by combining three already existing Agile methodologies DSDM, XP, and Scrum [11]. All of these are famous Agile methodologies and are widely used in software industry. The researchers further implemented this hybrid methodology on a real-time medium scale project in order to measure its accuracy. The results obtained are quite good. They further compared the results with the results of a project that is implemented using DSDM. A brief introduction of DXPRUM and DSDM is given here in order to understand these methodologies before comparing their results.

## 2.1. DXPRUM

DXPRUM is the proposed hybrid model and is a combination of three of the widely used Agile models named Dynamic Software Development Method (DSDM), Extreme Programming (XP) and Scrum. DXPRUM combines the best features of all the three models by removing their weaknesses. It combines the project management practices of Scrum with business-focused development approach of Dynamic Systems Development Method (DSDM) and by covering whole Software Development Life Cycle (SDLC) under the umbrella of engineering practices of Extreme Programming (XP). This makes DXPRUM a more powerful model for medium scale projects as it combines the strengths of the three chosen models. The best features of DXPRUM model includes management of software projects in a fixed time constraint with minimum possible resources and cost involved and also to work well with changing requirements. The defect rate of DXPRUM is also very low when compared with the three Agile models. All these positives result in a high-quality software product.

The DXPRUM have combined features of DSDM, XP and Scrum. It has pre-project and post-project phases of DSDM along with some SDLC features. The system backlog, sprint backlog, and DXPRUM increment features are chosen from Scrum. During whole SDLC, the different phases are covered under the umbrella of

engineering practices of XP that include coding standards, pair programming, refactoring, collective ownership of code and test driven development. Juxtaposition of all these makes DXPRUM an interesting model for software development organizations to adopt.

The DXPRUM model is shown in Figure 1. It starts with the pre-project phase in which a feasibility study of the software to be built is conducted. This feasibility study ensures whether the software product to be built may be completed within certain time constraints and budget. The pre-project phase is conducted before the project is officially started after which the first important phase of SDLC begins. This is the functional prototype phase where the initial design of the product is constructed. The input to this phase is the Software Requirements Specification (SRS) document. Then the system backlog is created where the requirements classification is done according to their priorities. The whole process is completed with the consultation of the stakeholders that involves DXPRUM expert, product owner, and DXPRUM production team. Then the product is divided into sprints where the high priority requirements are completed first. Then the main phases of SDLC are started. These include detailed design of the product, development and testing phases. These phases are conducted under the umbrella of XP practices. These include coding standards, pair programming, refactoring, collective ownership of code and test driven development.

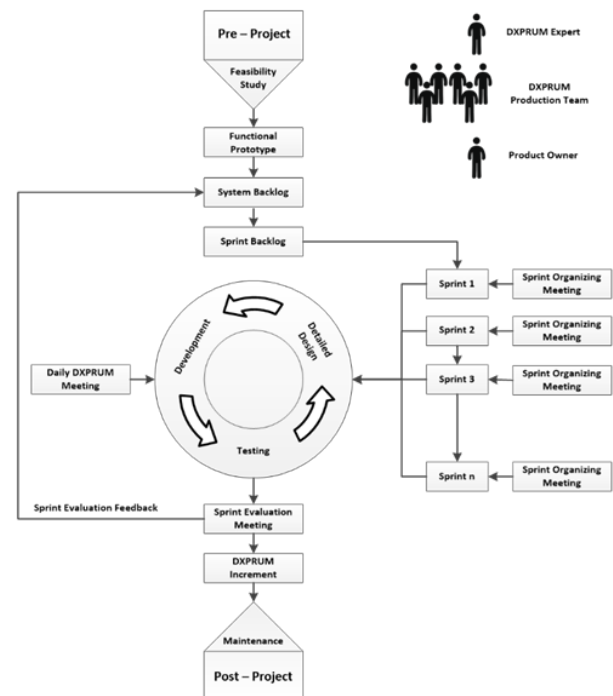


Fig. 1 DXPRUM Model [11]

A sprint evaluation meeting is conducted after the completion of each sprint. The meeting ensures whether all the requirements of that sprint are implemented or not. If there are any remaining or new requirements explored in that meeting these become part of the Sprint Evaluation Feedback (SEF) document. This document is the input to system backlog of next sprint. If all requirements are implemented and the product owner is satisfied from the sprint output then the sprint release is considered as DXPRUM increment. The final stage of the DXPRUM is the post-project phase, the one where maintenance of the project is done in future. If any bug is discovered in the product in future it is resolved in the maintenance phase.

The whole process of DXPRUM intends to cover all the aspects of SDLC that are necessary for the production of premium quality software with low cost and low defect rate. Each sprint cycle in DXPRUM is of 7-10 days long. A medium size project contains 4-8 of such sprints. Thus the duration of the project varies from 1.5 to 3 months. The sprint organizing meeting in DXPRUM is of 2-4 hours of duration. The daily DXPRUM meeting is of 30-45 minutes of duration. Once again the sprint evaluation meeting is of 2-4 hours of duration. The different roles of DXPRUM model are DXPRUM expert, product owner, and DXPRUM production team.

## 2.2. DSDM

Dynamic Systems Development Method is a framework for building high-quality business solutions. It is a well-organized framework that deals with the projects where there are strict time constraints. The Dynamic Systems Development Method used incremental prototyping and followed an iterative approach. Here each one of the iteration follows 80% rule in order to initiate the next iteration. The incremental prototyping allows parties to have a clear picture of the project throughout the software development lifecycle. The best aspect of Dynamic Systems Development Method is that it provides an environment where all interested parties involved in a project can cooperate and collaborate for successful completion of the project [12].

The Dynamic Systems Development Method is a modified version of Pareto Principle. In Pareto Principle, 80% functionality of a project is delivered in 20% of the time in order to deliver a complete 100% of the project. The remaining 20% of the functionality is deliberately left for later iterations. This strategy deals best with continuously changing requirements because we know that 100% of the requirements are not best known to the developers at the start of the projects and they kept on changing throughout the project life cycle. The Dynamic Systems Development Method is similar in nature to Scrum and XP. The only difference lies when there are

fixed time constraints. This is the main difference between Dynamic Systems Development Method and the other two models. In traditional software development methods, the time and resources involved in the project kept on changing all the time leaving functionality to be constant. But in Dynamic Systems Development Method, the time and resources remain constant and functionality kept on changing throughout the project life cycle. The process of Dynamic Systems Development Method consists of seven phases [13]. These include pre-project, feasibility study, business study, functional model iteration, design and build iteration, implementation and post-project phases. A comprehensive model of DSDM is shown in Figure 2.

The main benefits of DSDM include quick and in time delivery of the project with also an eye on reducing the cost of the project. It also involves self-organizing and collaborated teams. As the product is developed in prototypes, more requirements can be added after regular intervals. The feasibility study is done before the start of the project which reduces the chances of failure of the project. DSDM emphasizes on continuous collaboration between all parties. One of the main drawbacks of DSDM is that it did not work well with all type of projects. So the nature of the projects should first be analyzed in order to know and decide whether concepts and principles of DSDM, in this case, will help out or not. The process of DSDM also needs full commitment of the management. The main points not discussed in DSDM process consist of team size, the exact length of iterations and criticality of the system being developed. The different roles defined in DSDM are ambassador, visionary and advisor. There are also some other roles involved. The process of DSDM is used in more than 500 major companies in the world [14].

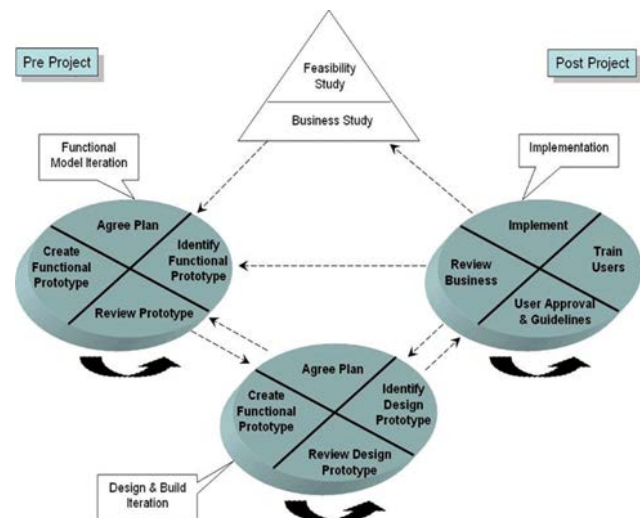


Fig. 2 Dynamic Software Development Method (DSDM) [11]

### 3. Results and Discussions

The DXPRUM case study was carried out in a software house in Multan, a city in southern Punjab province of Pakistan. The name of software house was DreamSpark. The software house has a team of experienced members having ample knowledge of application development along with higher degrees in computer science. The employees were already using Agile in software development. Furthermore, a one-week in-house training program was conducted in order for team members to be aware of DXPRUM model thoroughly. An online shopping cart system was implemented using DXPRUM model. There was 4 team member and 2 other members were performing the role of DXPRUM Expert and Project Manager. The case study consists of 4 releases with an overall 1.5 months of duration. The team was balanced in all aspects of software development. The development team ensured that the development is going on in the right direction under the supervision of DXPRUM Expert and Project Manager and by using the guidelines of proposed DXPRUM model. The results obtained were quite good and are shown in Table 1.

The DSDM case study was carried out in a British Telecom project and is written by “Ben Whittle” who was the project manager of the said project. It was further published in a book named “DSDM: Business Focused Development” written by “Jennifer Stapleton” [14]. British Telecom had already used DSDM in a number of their projects. The long working schedule is one of the major drawbacks of DSDM. The developers in DSDM are often burnout due to extensive and prolonged working hours. The case study used all the major techniques of DSDM and is also conducted with full facilitation support. The time estimation of the project was 12 weeks in length. But it was only completed in 7 weeks duration. The project consisted of 5 full-time team members. Throughout this project, the telephone and internet (skype) were used as a communication medium between customer and development team as the customer was in a remote location. The development team worked full time on the project. The project took 5 complete iterations and the duration of each iteration is 1 week. The remaining two weeks were used for Pre-project and Post-project phases (one week for each phase). The tools and techniques used in the project are shown in Table 1.

We have compared the results of DXPRUM case study with the results of DSDM case study. The comparison of both projects is shown in Table 1. The comparison is between two medium scale projects. We can see from the table that the number of iterations, number of team

members and the actual duration of the project are almost similar. One difference between two projects is the customer involvement. DXPRUM project has on-site customer while DSDM project has remote customer involvement. The second difference lies in a total number of lines of code while the third difference is between total post-release defects.

Table 1: Comparison of DXPRUM and DSDM

Sr No.	Parameter	DXPRUM	DSDM
1	Project Name	Online Shopping Cart	British Telecom
2	Project Size	Medium	Medium
3	No of Iterations	4	5
4	Estimated Duration (in Weeks)	7	12
5	Actual Duration (in Weeks)	6	7
6	Team Members	4	5
7	Customer satisfaction	High	Medium High
8	Customer Involvement	On Site	Remote
9	Programing Language	HTML, CSS, jquery, Ruby O Rails	HTML
10	Programing Environment	Dreamweaver	NEXTSTEP OO Development Environment
11	Other Tools	Eclipse, IBM Rational Rose	Devman, Code Reviewer
12	Database	MySQL	Oracle
13	Web Server	Apache	Webrex
14	Customer Satisfaction	92%	80%
15	Suitable to all Projects	Yes	No
16	Total Lines of Code (LOC)	54718	96024
17	Total Kilo Lines of Code (KLOC)	54.718	96.024
18	Post-Release Defects	13	94
19	Performance / Quality (Post-Release Defects / KLOC)	0.237	0.978

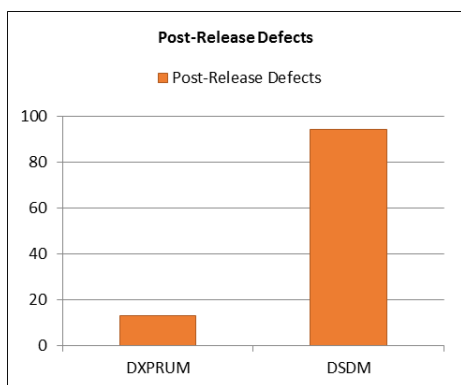


Fig. 3 Post-Release Defects

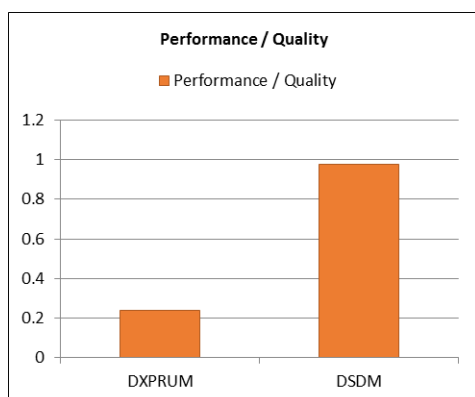


Fig. 4 Performance / Quality Graph

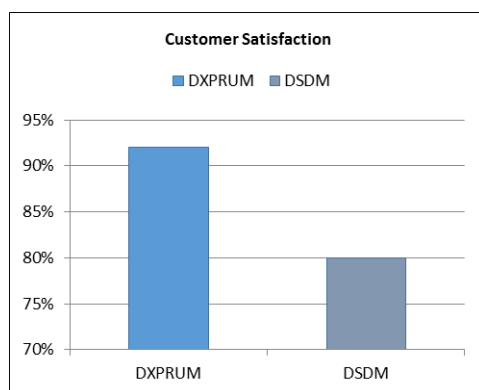


Fig. 5 Customer Satisfaction Graph

We can also see from table 1 that post-release defect rate of DXPRUM is also less as compared to DSDM. In DXPRUM the post-release defect occurs after every 4503 lines as compared to DSDM where the post-release defect occurs after every 1022 lines. As a result of it the overall performance/quality of the project is increased, which is 0.237 in DXPRUM as compared to 0.978 in DSDM. The less the number, in this case, is an indication of better quality. This parameter clearly proves our hypothesis

about DXPRUM model and shows that DXPRUM is much better than DSDM. Similarly, the customer satisfaction in DXPRUM model is about 92% while in DSDM it is about 80% when compared them with traditional software development methodologies. This makes a 12% increased satisfaction rate when DXPRUM model is preferred over DSDM model, and an overall 92% increased satisfaction rate when DXPRUM is compared with traditional software development methodologies. This parameter is obtained with the help of a survey. The figure 3, figure 4 and figure 5 shows the graphical representation of the parameters of post-release defects, performance/quality and customer satisfaction of DXPRUM and DSDM.

Table 2: Comparison with Traditional Methodologies

Sr No	Parameter	DXPRUM	DSDM
1	Improvement in Customer Satisfaction and On-time Delivery	43%	23%
2	Improved Organizational Skill for Management and Development Professional	85%	79%
3	Increased Process Predictability, Higher Maturity Level	65%	40%

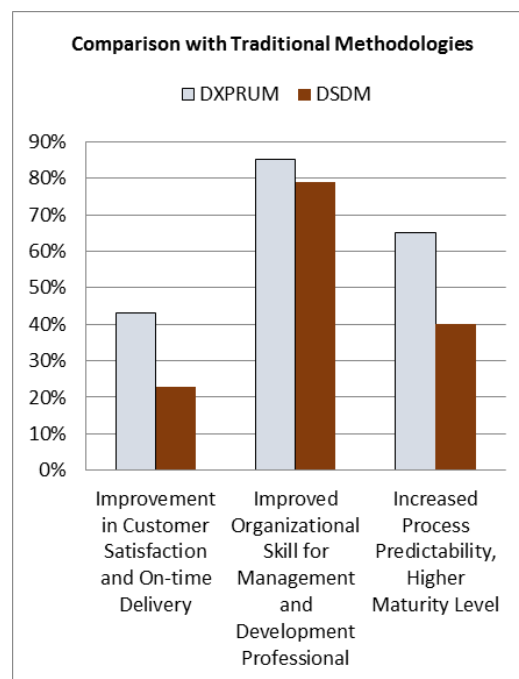


Fig. 6 Comparison with Traditional Methodologies

Table 2 shows the comparison of some of the key parameters of DXPRUM and DSDM with old traditional software development methodologies. The three parameters compared are customer satisfaction and on-time delivery, improved organizational skills of management and development professionals and increased process predictability and higher maturity level. The displayed results clearly showed that both DXPRUM and DSDM are much better than old traditional software development methodologies. This comparison is also shown in the form of a graph in Figure 6.

#### 4. Conclusion

DSDM is an already existing popular Agile methodology, while DXPRUM is a new methodology and is the combination of three Agile methodologies (DSDM, XP and Scrum). The DXPRUM needs some time to get mature. We have implemented DXPRUM model on a real time project and compared the obtained results with an already existing case study of a British Telecom project that used DSDM for development. The comparison is shown in the form of tables and figures. The results clearly showed that DXPRUM performed better and seems to be a more powerful model. By implementing DXPRUM we are able to deliver more high quality and cost effective software with the minimum possible amount of time and low cost involved. The defect rate of this softwares is also low as compared to DSDM. This makes DXPRUM a more compact, concise and user friendly model and the one that needs lots of attention in near future.

#### References

- [1] Edeki C., "Agile Software Development Methodology," *European Journal of Mathematics and Computer Science*, vol. 2, no. 1, pp. 22-27, ISSN: 2056:600X, 2015.
- [2] "Software and Services in Large Enterprises," *Business Technographics, Forrester Research*, 2005. [www.forrester.com/Research/Document/Excerpt/0,7211,38659,00.html](http://www.forrester.com/Research/Document/Excerpt/0,7211,38659,00.html).
- [3] Boehm B., "Get Ready for Agile Methods, with Care," *Computer*, vol. 35, no. 1, pp. 64-69, 2002.
- [4] Nerur S., and Balijepally V., "Theoretical Reflections on Agile Development Methodologies," *Comm. ACM*, vol. 50, no. 3, pp. 79-83, 2007.
- [5] Mushtaq Z., and Qureshi, M. R. J., "Novel Hybrid Model: Integrating Scrum and XP," *International Journal of Information Technology and Computer Science (IJITCS)*, vol. 4, no. 6, pp. 39, 2012.
- [6] Bashir M. S., and Qureshi M. R. J., "Hybrid Software Development Approach for Small to Medium Scale Projects: Rup, Xp & Scrum", *Cell*, 966, 536474921. 2012.
- [7] Qureshi M., "Empirical Evaluation of the Proposed eXSCRUM Model: Results of a Case Study". arXiv preprint arXiv:1202.2513, 2012.
- [8] Rasool G., Aftab S., Hussain S., and Streitferdt D., "eXRUP: A Hybrid Software Development Model for Small to Medium Scale Projects," *Journal of Software Engineering and Applications*, vol. 6, no. 9, pp. 446, 2013.
- [9] Malhotra C., and Chug A., "IXSCRUM-A Framework Combining Scrum and XP," *International Journal of Scientific & Engineering Research*, vol. 4, no. 7, 2013.
- [10] Ahmad G., Soomro T. R., and Brohi M. N. "XSR: Novel Hybrid Software Development Model (Integrating XP, Scrum & RUP)". *International Journal of Soft Computing and Engineering (IJSCE)*, vol. 2, no. 3, 2014.
- [11] Fahad M., Qadri S., Muhammad S. S., and Husnain M., "Software Quality Assurance of Medium Scale Projects by using DXPRUM Methodology," *International Journal of Natural and Engineering Sciences (IJNES'14)*, (Turkey), vol. 8, no. 1, pp. 42-48, 2014.
- [12] Stapleton J., "DSDM: Dynamic Systems Development Method," In *Proc. of Int. Conf. of Technology of Object-Oriented Languages and Systems*, (Nancy, France), pp. 406, ISBN: 978-0-7695-0275-5, 1999.
- [13] Voigt B. J. J., "Dynamic Systems Development Method," MS Thesis, Department of Information Technology, University of Zurich, (Zurich, Switzerland), 2004.
- [14] Stapleton J., "DSDM: Business Focused Development," 2nd Ed, Addison Wesley Professional, The DSDM Consortium, (London, UK), pp. 145-163, ISBN: 0-321-11224-5, 2003.