The Nucleus ***53****, No. 3 (2016) 180-187*

www.thenucleuspak.org.pk

The Nucleus

ISSN 0029-5698 (Print)
ISSN 2306-6539 (Online)

A Hybrid Software Architecture Evaluation Method for Dynamic System Development Method

R. Awan[1*], S.S. Muhammad[1], M.A. Fahiem[2] and S. Awan[1]

[1]*Department of Computer Science, Virtual University of Pakistan, Jinnah Campus, Lahore, Pakistan*

[2]*Department of Computer Science, Lahore College for Women University, Lahore, Pakistan*

ms110400102@gmail.com ; syed@vu.edu.pk ; abuzar@lcwu.edu.pk, sawan@vu.edu.pk

ARTICLE INFO

Article history :
Received : 06 November, 2015
Revised : 17 August, 2016
Accepted : 22 August, 2016

Keywords:
Agile Process Models
SAEM
DSDM
ADD
ARID
QAW

ABSTRACT

Quality is an important paradigm while delivering software. Past traditional models have many problems. Agile process models overcome these problems, but these models are facing many challenges. The main challenge is the absence of proper Software Architecture Evaluation Method (SAEM) for agile models. It is essential to improve quality of agile models because these methods are lacking quality requirement, well defined software architecture and verified design. Without proper evaluation, these models suffer from severe quality and maintenance issues. Architecture evaluation is known as a standard to evaluate the quality of product. This study focuses on the development of a hybrid SAEM for agile process model. Dynamic System Development Method (DSDM) is a framework of agile methodology. This framework delivers a quality product in a short time. It is very important to improve quality of DSDM phases. Pre-project, project life-cycle and post-project are the phases of DSDM. It is required to apply a hybrid SAEM in phases of DSDM. By applying a hybrid SAEM on DSDM, the quality of DSDM phases may be improved. This improvement may be in term of quality attributes which are well defined in early life cycle. Furthermore, the quality attribute requirements are best satisfied due to well-formed software architecture design. A survey has been conducted in the software industry to validate this model.

1. Introduction

Agile methods are a group of software development methods that deliver working software in minimum time span. According to [1-2] agile methods have number of benefits as they support iteration and constant interaction between developer and customer. Due to minimum documentation these are lightweight. Usability is one of quality factors that may be addressed in agile methods.

One problem with agile methods is that explicit qualitative and quantitative measurements and metrics are avoided. Sometimes agile methods cannot cope with insufficiency in the product [3]. DSDM works in an iterative and incremental manner with constant user involvement. At the start of project, DSDM tasks are arranged according to their importance. DSDM have benefit of ease in adopting, constant user support and development tool support. Pre-project stage deals with the identification of suitable projects and budgeting. Business study is concerned with the business area definition, project scope, high level functional and non-functional requirements, system functionality, architecture, and maintainability objectives. Functional model iteration deals with requirement refinement and prioritization, identification of non-functional requirements and planning for implementation. Design and development iteration deals with implementation of identified requirements, creation of identified prototype and review of the prototype. Implementation deals with product delivery. Performance of implemented system is dealt in post project activities.

Software architecture evaluation is a vital approach to develop better quality product in the software engineering area. Without appropriate evaluation, software artifact suffers from severe quality drawbacks. If a hybrid SAEM will be applied on DSDM, the quality will improve in terms of attribute requirements, design constraints, functional requirements and suitable design.

Following are objectives which are addressed in this study:

- To identify problems in traditional software development models?
- What are reasons to accept or reject agile methods?
- Problems faced by software industry while working with DSDM.
- Do the organizations use software architecture evaluation methods?
- Effectiveness of a hybrid SAEM in solving the problems in DSDM.

* Corresponding author

SAEM assists developers in software architecture creation that can accomplish the required goals of the system [4]. The aim of this study is to introduce a hybrid SAEM for DSDM phases. The proposed model composes of Quality Attribute Workshop (QAW), Attribute Driven Design (ADD) and Active Review for Intermediate Design (ARID).

The objective of study is to improve DSDM phases and apply SAEM on it. The aim is to get a better understanding of SAEM, design a different SAEM and find compatibility between QAW, ARID, ADD and DSDM phases. In the past, there is work on DSDM and SAEM but SAEM is not applied on DSDM.

The rest of the structure of this paper is as follows; Section 2 presents the related work, section 3 elaborates the proposed work, section 4 describes the results and discussion. Finally, section 5 concludes the paper and section 6 provides future direction.

2. Related Works

Rao et al. elaborated that software development is a wide field that delivers the product in a quicker and cheaper way [5]. The most common methods discussed in this paper are extreme programming, DSDM, SCRUM and Crystal [6]. The aim of these methods is quick delivery of software with customer satisfaction and minimum number of iterations. The authors discussed principle and reason for using agile methods. Pros and cons of above mentioned agile methods were discussed. One of the main reasons for using DSDM is that it is technique-independent process and flexible because of requirement growth. One problem is continuous user involvement which may not be possible at every time in the project. The authors did not provide solution to this problem.

Jyothi and Rao examined that DSDM is a method developed by a dedicated consortium in the UK released in 1994 [7]. The DSDM is based upon rapid application development (RAD) in which software development puts less emphasis on planning and more emphasis on development. The basic idea of this method is to adjust functionality within limited time and resources. Architecture transformation and other architecture evaluation techniques are defined to overcome the problems concerned with non-functional requirements [8].

Buchgeher and Weinreich explained that software architecture evaluation plays an important role in software development [9]. It is constructive and supportive process for design and implementation. Best time to evaluate architecture is before its implementation. The late evaluation can be applied when development team starts making decisions. The architecture evaluation makes it sure that the system will meet its quality goals [10]. SAEM when applied in DSDM phases, resolves conflicting requirements, architecture design decisions, consistency checking, early problem detection and cost issues. The cost and benefits are two motivations achieved by Software Architecture Evaluation [11]. SAEM's have standard steps for any development paradigm.

Sohaib and Khan explained that software plays a major role in the industry; therefore new methodologies are introduced by the software engineering community [12]. After iteration, team discusses problems and proposes their solutions. The term agility concentrates on software quality. In traditional development methods, projects fail due to huge time period. Agile methods overcome these issues via quick development. Different software quality factors were explored in DSDM like efficiency, integrity, ease of use, maintainability, testability and flexibility. Integrity is architecture driven; and re-usability is achieved by Object Oriented Unified Modelling Language (OOUML) patterns [13]. Agile methods put more pressure on developers to write code that can easily be understood by other members in the team. Also, there are other quality factors like quality attribute requirements, verified design and usability that need to be addressed by agile.

Kanwal et al. examined Feature Driven Development, (FDD) phases and introduced a hybrid SAEM. This hybrid model composes of QAW, ARID and Architecture Tradeoff Analysis Method (ATAM). For phase 1 of the FDD, both functional and non-functional requirements gathering is executed in parallel so QAW is better because it produce, prioritize and purify the quality attribute before architecture is finalized. For phase 2, both ARID and ATAM are applied [14]. Kazman et al. discussed the features of ATAM which are utility trees, sensitivity points and trade-offs are helpful in designing a proper architecture [15].

Aydin et al. explained that purpose of using DSDM is to make sure that the project should be developed in a short time span to attain product and process improvement [16]. In the UK and in Benelux countries, DSDM is supported by a consortium of 600 organizations. The DSDM method stressed upon the concept of suitability and adaptability. Software architecture evaluation makes sure that the selected software architecture will meet both functional and non-functional quality requirements. Software architecture evaluation increase understanding and system documentation, problem detection and improve organization learning.

Abrahamsson et al. promoted the idea to fix time and resources and amended functionality. This work gave more importance to process quality than quantity [17]. Coyle and Conboy [18] stated that there is always risk involved either the change is for business development or

operational. However, this paper focused on risk management only in DSDM. The team focuses more on risk management in DSDM than other agile methods.

Lassing *et al.* elaborated the significance of software architecture. The quality of the system can be better predicted by analysis of software architecture. The aim of software architecture evaluation is to maintain cost, risks. Flexibility is attribute which can be achieved by software architecture analysis. It helps system to adopt changes [19]. Shaw stated that conflicting requirements can be settled down in better manner by taking design decisions. Early design decision helps in problem solving and express design in better manner [20].

Nerur *et al.* elaborated that software methodologies are changing due to changing technology and varying user demand. The traditional development methodologies cannot adjust dynamic changing business environment. Agile development methodologies evolve to overcome problems in traditional approaches [21].

3. Proposed Model

The flow of project starts from the pre-project stage. The feasibility study deals with assessment of feasibility of application. The next stage is business study where project activities, functional, non-functional requirements, system architecture and maintenance objectives are outlined. Quality attributes workshop is best suitable SAEM as it deals with analysis and refinement of requirement from quality point of view.

The next stage is functional model iteration as this phase deals with identification, creation and review of functional prototype so no SAEM suits there. The next phase is design and builds iteration. In design and build iteration the first module is identification of design prototype, here ADD is a suitable activity as in the identification of the design prototype module are chosen for decomposition, architectural drivers are selected, architectural plan is chosen, the module is instantiated and functionality is allocated via use cases, interface for child module are defined, use case is verified and considered as a constraint for child models. By creating designs prototype phase, software architecture design is created. Lastly design prototype is reviewed .ARID deals with design process review and explores the design in more detail. Design presentation prepared and all possible scenario are identified that design cover. In this approach design is evaluated from different perspectives.

The last stage is implementation where user approval, user training, implementation and business review performed. As no SAEM suit there so it left as it is. Table 3 shows the detailed view of the proposed model. The pictorial representation of proposed work is shown in Fig. 5.

4. Results and Discussion

This section presents results from survey data analysis. The results are summarized in tables and presented in the form of different graphical figures. The proposed model is tested against survey results. Industry personnel describe their problems faced while using traditional models. They explained reasons for rejecting agile methods and problems while working with DSDM. Positive outcomes were found in survey as industry personnel showed their willingness to use proposed model. They found it effective in perception of quality requirement, better and approved design.

4.1 Reason for Tried and Rejected Agile Method

It is observed that many software organizations used agile methods, but eventually rejected them. When reasons for rejection were asked, the results are shown below in Table 1.

Table 1: Reason for tried and rejected agile method

Problem	No of reason
There is a lack of emphasis on necessary designing and documentation.	15
In case of some software deliverables, especially the large ones, it is difficult to assess the effort required at the beginning of the software development life cycle.	20
The project can easily get taken off track if the customer representative is not clear what the final outcome that they want.	28
None	37

4.2 Problems Faced While Adopting DSDM

Questions were asked regarding to problems while adopting DSDM. Mostly software analysts and developers agreed with the issue that it is not suitable for large projects because of short timeline. The membership constraint is another problem. Software requirements must be fully known before applying DSDM in majority of complex systems. It is very tricky to know all requirements in advance. The module cannot be delayed. Table 2 show details of problems occurred in DSDM. 4.3 Proposed Model Overcomes Problems in DSDM

4.3 Proposed Model Overcomes Problems in DSDM

Software development team consisting of analysts and project manager gave feedback about proposed model. The figure 1 shows the details. Most of people agreed that proposed model overcome problems incorporated in DSDM.

Table 2: Problems faced while adopting DSDM

Issues	Percentage
Not suitable for large project	12
Membership Constraint	8
Requirements Must Be Fully Specified	10
Computationally Complex Systems	5
All Requirements Must Be Known Before System Build	9
All Features Must Be Implemented And Delivered For Smooth Working Of System	11
All Above	45

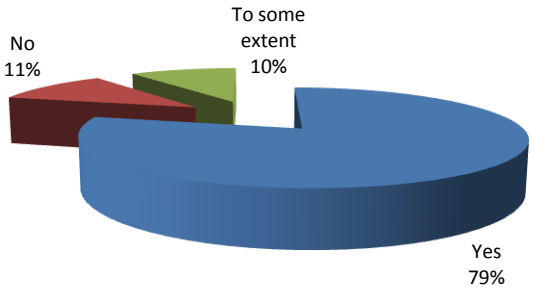

Fig. 1: Proposed model overcome problems in DSDM

4.4 *Percentage of Requirement Problems Overcome While Using Proposed Model*

What percentage the requirement problems can be overcome by the proposed model. The results are shown in Fig. 2.

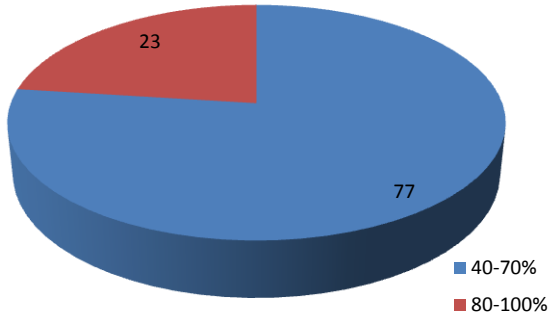

Fig. 2: Percent of requirement problems overcome while using proposed model

4.5 *Output of Proposed DSDM in Tightly Scheduled Projects*

Fig. 4 shows the behavior of proposed DSDM in projects where time line is strict. The majority of responses are between range of high and medium.

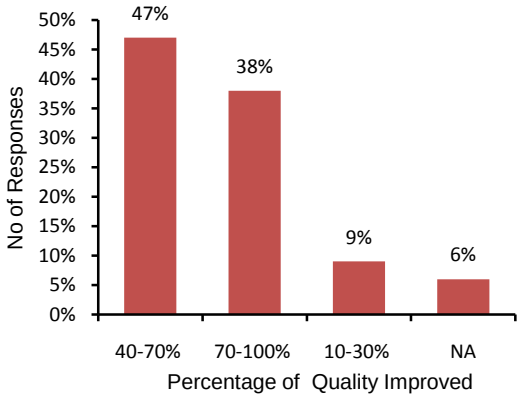

Fig. 3: Improvements in software quality by using prop1osed model

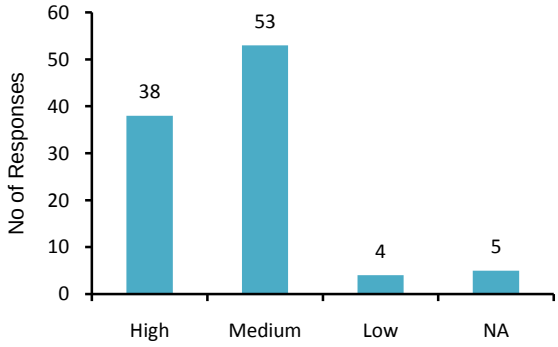

Fig. 4: Output of DSDM in tightly scheduled projects

4.6 *Finalization of Quality Attribute Requirements by Applying Quality Attribute Workshop*

How much quality attribute requirements are finalized by applying quality attribute workshop on DSDM phases? Large number of people agrees that 40-60% requirements are finalized by applying quality attribute work shop. A simple column chart is presented in Fig. 6.

4.7 *Organizations Trend to Use the Proposed Development Model In Future*

Most of software organizations are willing to use the proposed model in their future project as shown in Fig. 7.

4.8 *Effect of Proposed Model for DSDM Speed*

The effect of the proposed model on DSDM speed was observed. Graphical representation is shown in Fig. 8.

4.9 *The Proposed DSDM Satisfies Which Factor Most Efficiently*

Different factors like management expectation, customer satisfaction, time and cost improvement, better business solution were evaluated. Customer satisfaction is factor which is satisfied mostly by applying proposed model. Results are summarized in Fig. 9.

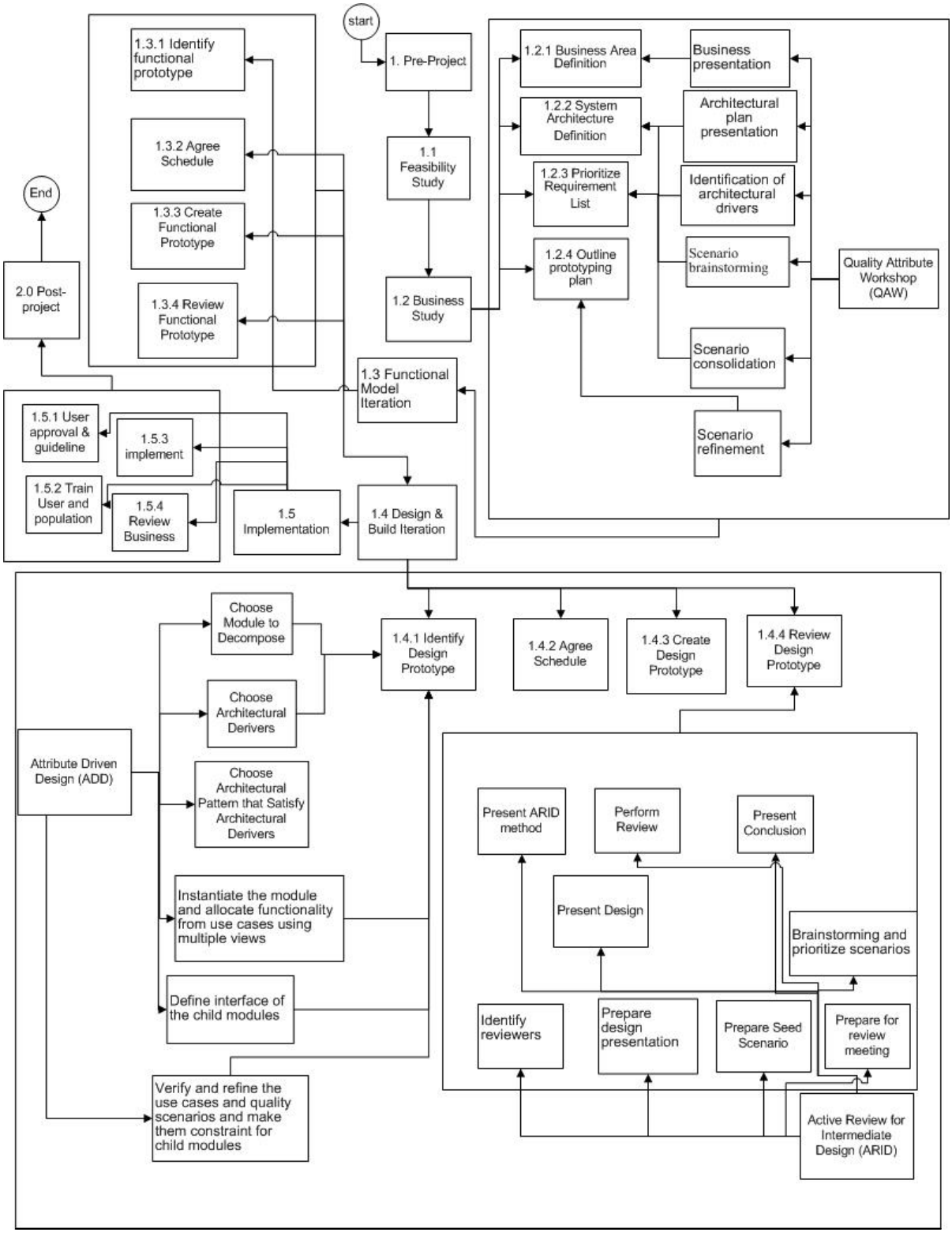

Fig. 5: Mapping of architecture evaluation activities on DSDM life cycle phases

Table 3: DSDM and architecture evaluation activities

Phases of DSDM			Architecture Evaluation Activity	Model
Phase	Activity	Sub Activity		
Study	Feasibility Study	Feasibility Prototype	N/A	None
		Feasibility Report		
		Global Outline Plan		
		Development Plan		
		Risk Log		
	Business Study	Business area definition	Business presentation	QAW
		System architecture definition	Architectural plan presentation	
			Identification of architectural drivers	
		Prioritize requirement list	Scenario brainstorming Scenario consolidation Scenario prioritization	
		Outline prototyping plan	Scenario refinement	
Functional model iteration	Identify Functional Prototype	Functional model	N/A	None
	Agree a schedule	Functional prototype		
	Create functional prototype			
	Review functional prototype	Functional prototype review document		
Design and build iteration	Identify design prototype	Implementation strategy	Choose the module to decompose	ADD
			Choose architectural drivers	
			Choose the architectural pattern that satisfies architectural drivers	
			Instantiate the module and allocate functionality from use cases using multiple views	
			Define interface of the child modules	
			Verify and refine the use cases and quality scenarios and make them constraint for child modules	
	Agree a schedule		N/A	None
	Create a design prototype	Design prototype	Software architecture design	
	Review design prototype	User documentation or test record development	Identify reviewers	ARID
			Prepare design presentation	
			Prepare seed scenarios	
			Prepare for review meeting	
			Present ARID method	
			Present design	
			Brainstorming and Prioritize scenarios	
			Perform review	
			Present conclusion	
Implementation	User approval and guidelines	User approval	N/A	None
	Train user	Train user population		
	Implement	Delivered system		
	Review business	Project review document		

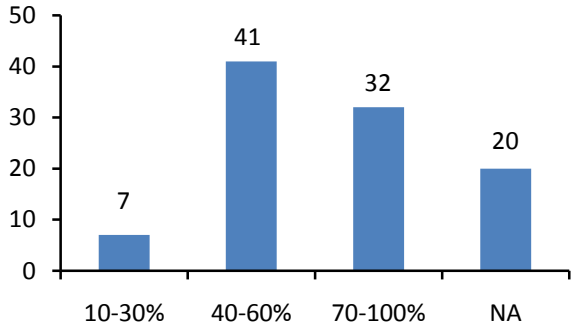

Fig. 6: Finalization of quality attribute requirements by applying quality attribute workshop

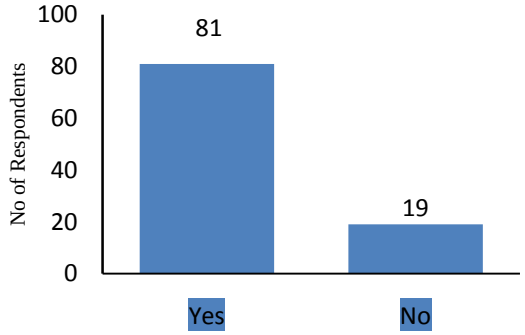

Fig. 7: Use of proposed model in future

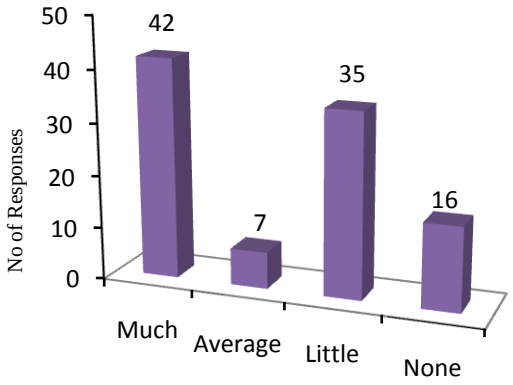

Fig. 8: Effect of proposed model for DSDM speed

The proposed work is different from previous work in the sense that it applies software architecture evaluation activities on DSDM - an agile process model. The life cycle phases of DSDM which are related to requirement analysis, early confirmation of the functional and non-functional requirements, software design and implementation phases are improved in term of quality and a mature system design comes to the surface. Well-formed software architecture is established. A critical review of all these activities eliminates the need of further bugs. The hybrid SAEM minimize the need of testing as maximum chances of errors are eliminated during early life cycle phases. The time and resources are saved.

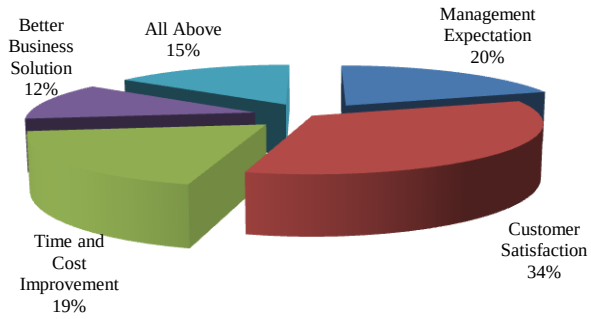

Fig. 9: DSDM satisfied which factor efficiently

5. Conclusion

Agile methods mitigate issues in traditional process models, but these methods suffer from severe quality and maintenance drawbacks. Software architecture evaluation is a vital approach to develop a better quality product. A hybrid SAEM is applied on DSDM-an agile process model to improve its quality. The life cycle phases of DSDM need much improvement in terms of functional, non-functional requirements, quality attributes and suitable design. Software architecture evaluation is a standard which proved to be helpful to achieve these quality requirements. To achieve this hybrid SAEM is proposed which is composed of QAW, ADD and ARID. The survey, conducted from market shows the results about traditional models and their problems, problems related to agile methods, problems concerned with DSDM, software architecture evaluation methods used by the software industry, problems sort out and solved by the proposed model. All these questions are answered by software development professionals. The results of all above questions are summarized in result and discussion section. The results show that software industry like the proposed model. By the application of our model requirements are more refined and software is developed with all quality attributes with minimum cost. This is a major distinction of this study as in past organizations are only using DSDM where systems are not up to the mark with many loophole in requirement and design.

6. Future Work

There is need of further exploration of other software architecture evaluation methods to make them suitable for any other flavor of agile process models.

When it was asked about un-addressed problems 23% people answered that it could not solve problem of large scale integration. The graph in figure 3 summarizes the results.

The quality and long term working of software heavily depends upon its architecture. The emphasis of such evaluation is to improve quality of software in cost and schedule perspective.

References

[1] L. Williams, "A survey of agile development methodologies", NC State University, North Carolina, United State, pp: 209-227, 2007.

[2] A. Bagel and N. Nagappan, "Usage and perceptions of agile software development in an industrial context: An exploratory study", Empirical Software Engineering and Measurement: Proc. of the First Int. Symp. on Empirical Software Engineering and Measurement, ESEM 2007 WA, USA, Madrid: IEEE, pp. 255-264, 20-21 Sept., 2007, Available: IEEE Xplore, http://www.ieee.org.

[3] B. Boehm, "A survey of agile development methodologies", NC State University.2007 (North Carolina, USA), pp. 209-227.

[4] F. Martensson, "Software architecture quality evaluation-approaches in an industrial context", Blekinge Institute of Technology. (Karlskrona, Sweden). ISSN: 1650- 2140, 2006.

[5] K. N. Rao, , G. K. Naidu and P. Chakka, "A Study of the agile software development methods, applicability and implications in industry, International Journal of Software Engineering and Its Applications, ISSN: 1738-9984 (Tasmania, Australia), vol. 5, no. 26, pp. 35-45, 2011

[6] M. Cristal, D. Wildt and R. Prikladnicki, "Usage of scrum practices within a global company", Proc. of Int. Conf. on Global Software Engg., ICGSE 2008, pp. 222-226, 17-20 Aug. 2008, Bangalore, India,. Available: IEEE Xplore, http://www.ieee.org.

[7] V. E. Jyothi and K. N. Rao, "Effective Implementation of Agile Practices Ingenious and Organized Theoretical Framework", International Journal of Advanced Computer Science and Applications, vol. 2, no. 3, pp: 41-48, March 2011.

[8] P.O. Bengtsson and J. Bosch, "Scenario-based software architecture re-engineering", Proc. of 5th Int. Conf. on Software Reuse, BC. Victoria, IEEE, pp. 308-317, 2-5 June 1998, available: IEEE Xplore, http://www.ieee.org.

[9] Buchgeher, G. and R. Weinreich. "An approach for combining model-based and scenario-based software architecture analysis", Proc. of 5th Int. Conf. on Software Engineering Advances (ICSEA'10). (Nice, France). pp. 141-148. ISBN: 978-1-4244-7788-3. 2010.

[10] P. Clements, R. Kazman and M. Klein, "Evaluating software architectures: methods and case studies", Boston: London, Reading: Addison-Wesley, 1st Edition, 2001.

[11] G. Abowd, L. Bass, P. Clements, R. Kazman, L. Northrop and A. Zaremski, "Recommended best industrial practice for software architecture evaluation", Soft Eng. Inst., Carnegie Mellon Univ., PA, USA, Tech. Rep. CMU/SEI-96-TR-025, pp. 1-33, Jan 1997.

[12] O. Sohaib and K. Khan, "The Role of software quality in agile software development methodlogies", Technical Forces, Journal of Engg. and Sci., Karachi, Pakistan, pp. 20-25, 2010.

[13] M. Fontoura, P. Wolfgang, and B. Rumpe, "UML-F: A modeling language for object-oriented frameworks", Proc. of the 14th Europ. Conf. on Object-Oriented Programming, ECOOP 2000, London, UK, Verlag: Springer, pp. 63-82, 2000.

[14] F. Kanwal, K. Junaid and M. A. Fahiem, "A hybrid software architecture evaluation method for FDD – An agile process model", Proc. of Int. Conf. on Computational Intelligence and Software Engg., CiSE 2010, Wuhan, China, pp. 1-5, 10-12 Dec., 2010.

[15] R. Kazman, M. Barbacci, M. Klein, S. J. Carriere and S. G. Woods, "Experience with performing architecture tradeoff analysis", Proc. of the 21st Int. Conf. on Software Engg., ICSE 1999, CA, USA, 16-22 May, 1999, pp. 54-64.

[16] M. N .Aydin, F. Harmsen, K. V. Slooten and R. A. Stagwee, "An agile information systems development method in use", Turk J. Elec. Engg., Ankara, Turkey, vol. 12, pp.127-138, 2004.

[17] P. Abrahamsson, J. Warsta, M. T. Siponen and J. Ronkainen, "New directions on agile methods: A comparative analysis", Proc. of 25th Int. Conf. On Software Engineering, IEEE 2003, CA, USA, pp. 244-254, May 3-5, 2003.

[18] S. Coyle and K. Conboy, "A case study for risk management in Agile Systems Development", Information Systems: Proc. of 17th Europ. Conf. on Inf. Systems, Verona, Italy, May 8, 2009, S. Newell, EA. Whitley, N. Pouloudi, J. Wareham, L. Mathiassen, eds, pp. 2567-257, 2009.

[19] N. Lassing, D. Rijsenbrij and H. V. Vliet,. "The goal of software architecture analysis: confidence building or risk assessment", Proc. of 1st BeNeLux Conf. on Software Architecture. (Amsterdam, Netherlands), pp. 47-57, 1999a.

[20] Shaw, M. "Comparing architectural design styles. IEEE Software". (CA, USA).vol. 12, no 6, pp: 27-41. ISSN: 0740-7459.1995.

[21] S. Nerur, R. Mahapatra and G. Mangalaraj, "Challenges of migrating to agile methodologies", Communications of the ACM (NY, USA), vol. 48, no. 5, pp. 72-78, 2005.