An Empirical Study of Heavy weight and Light weight Software Development Process Models

Manisha Uniyal[1], Arvind Kalia[2]

Research Scholar, Department of Computer Science, Summer-Hill, Shimla, India [1]

Professor, Department of Computer Science, Summer-Hill, Shimla, India [2]

Abstract: Selection of the suitable software development process model is not an easy task for the development of software project. A project Manager can use different criteria to select a right model for the development of software project as per requirement of the project. The selection of one model over the others is driven by Project size, Quality, Team size, Budget, Time, criticality of the project and a lot of other factors. A questionnaire was conducted; feedback from software professionals also gathered to evaluate which model has a better success rate for different sizes of software project.

I. INTRODUCTION

Software has a great influence in modern society that is why it plays an important role for more than 60 years. Software development was started off as an untidy activity often mentioned as a code and fix model. The software was written without much of a plan. This suited well for only small projects but as the size of the software grew it became more complicated to add new features and bugs were harder to fix. This style of software development was used for many years until an alternative was introduces.

Traditional models are plan driven in which work starts with gathering of requirements and documentation of complete set of requirements.

Heavyweight models are traditional models such as waterfall, prototype, incremental and spiral. These models are considered as heavyweight because it involves heavy documentation. Some practitioners found this process poses difficulties when new changes need to be implemented. As a result agile model was introduces along with its several approaches such as extreme programming (XP), Scrum and Crystal method. These methodologies are considered as lightweight because it involves light documentation.

Each model has its own advantages and disadvantages, and each software development process model may provide better functionalities in one situation than in another. Then the challenge is to decide which model should be selected to provide a particular set of functionalities under certain circumstances [7].

However, from different studies it has been find that there is no universal life cycle model, which can be considered in all situations [6].

Software developers, taking into account its goals and the size of a project, and have a number of well-established software development process models to choose from. Therefore, even though there are number of models and each software Development Company adopts the best-suited model, which facilitates the development process and boosts the productivity of its team members.

II. LITERATURE REVIEW

Qureshi [9] provided a brief overview of "an improved XP software development process model" The concept of agile process models has attained great popularity in software (SW) development community in last few years. Agile models promote fast development. Fast development has certain drawbacks, such as weak documentation and performance for medium and large development projects. Extreme Programming (XP) process model was modified and evaluated on the basis of different studies.

Subramaniam [12] conducted a comparative study of different agile methods on the basis of different features. More important, why should we learn about these different methodologies instead of simply focusing on one? There is no one shoe that fits all. To be effective they take the best of different approaches and apply to their projects base on their specific needs. **Abrahamsson et.al [8]** proposes a definition and a classification of agile software development approaches. Then ten software development methods were analysed against the different criteria. Finally these methods were compared; their similarities and differences were also highlighted on the basis of existing literature. **Cohen et.al [2]** Describes and compares some of the more popular agile methods,

provides a guide for deciding where an agile approach is applicable, and lists common criticisms. It summarizes empirical studies, anecdotal reports, and concludes with an analysis of various agile methods. **Saeeda et. al [10]** provided a brief overview of agile scalability for large scale projects. On the basis of literature review current research problems of agile scalability for large projects were identified. All the results were gathered will be summarized statistically based on these findings. The purpose of this study was to know agile techniques in detail, finding and highlighting its restrictions for large size projects.

Khan et.al [5] gave a comparative study of heavyweight and lightweight models on the basis of strength and weakness of two opposing methodologies. Some criteria were also mentioned that will help the project managers to choose the best model for the development of the software project. **Hawrysh [4]** state that one model cannot work for different projects. But according to specific nature of the software project developers can select the best model. **Mahanti [11]** present the results of the survey on factors affecting the choice of software life cycle models in the software industry conducted with software professionals in a few Indian software companies. **Dora et.al [3]** provided a comparative study of heavyweight and lightweight models on the basis of different studies and after study it was concluded that agile excels traditional models. **Mishra [7]** gave a comparative study of various life cycle models in different scenarios on the basis of certain features like user involvement, risk involvement, cost, time etc. **Maheshwari [6]** conducted a comparative study of various process models on the basis of various factors. Here the performance of each model was calculated on the behalf of various features.

III. NEED OF STUDY

There are various software development process models widely used for developing software. Software developing models play an important role for developing the software in a systematic manner such that it will be delivered within the time, budget and should also have proper quality. Each model has its advantages and disadvantages according to which it is decided that which model should be implemented under which conditions. For this various software development process models needs to be evaluated.

IV. OBJECTIVE

The main objective of the study was to analyse the success rate of models depending on the size and type of the project.

V. RESEARCH METHODOLOGY

An exhaustive review of literature was performed to study the various models used for software development. A survey was conducted among software professionals in

various software companies located in North India. The data was collected through well-structured questionnaire from different software companies. Information regarding their opinion on agile and heavyweight models was also collected to extract as much information as possible. The percentage analysis was done to reach the conclusion.

VI. QUESTIONNAIRE FORMAT

The items of the questionnaire were framed in such a way that it should be clear to the respondents and can answer the questions easily and quickly. The questionnaire was divided into three sections:

1. **Organization Characteristics:** These questions deal with the type and size of the organization**.**
2. **Software Development Items:** This section of the questionnaire emphasized on software development items. This was further categorized into four sections. These sections included general questions, quality questions, time questions, cost questions. Respondents provided their feedback on heavyweight models, lightweight models methodologies, project cost, software quality and development time.
3. **Additional Information:** Some of the additional questions were asked to get the other related information.

VII. RESULTS AND ANALYSIS

The items of the questionnaire were analysed and the results are discussed below with the help of tables and graphs.

1. Organization Characteristics and Knowledge:
More than 80% of the respondents were from information technology field, and the others were from telecommunication area. Furthermore, out of 18 companies, 80% respondents were from the companies having full time employees between 10 to 100 and remaining 20% of the respondents were from the companies over 300 full time employees.
54% of the respondents rate their knowledge of heavyweight and lightweight models as average. 32% were of the opinion that their knowledge about heavyweight and lightweight models is extensive. 9% having very extensive knowledge and the remaining respondents with limited knowledge were given lower rating to some questions because of their limited knowledge for this purpose.

2. Model used as per the Size of the Project: It is evident from Table. 1 that 62% of the respondents were using agile, 14% prefer prototype model and 24% support waterfall model for the development of small sized software projects. Agile proved to be good for small sized projects because it requires less people that can fit in a single room. Whereas for medium scale projects 45% of the respondents were using waterfall model, 25% prefer spiral model, 14% with prototype, 12% support agile

model and 4% favour incremental model, and for large scale projects 50% respondents were using waterfall, 32% prefer agile, 12% with spiral and 4% support prototype.

TABLEI FREQUENTLY USED MODELS AS PER THE SIZE OF THE PROJECT

Values in %

Models / Size	Water fall	Proto type	Agi-le	Incre mental	spiral
Small Scale	24	14	62	0	0
Medium Scale	45	14	12	4	25
Large Scale	50	4	32	0	12

Agile relay on communication, as project size increases, team size will also increase. It can cause distribution of team and thus communication breakdown would be occurred within the team members. It is clear from analysis that software companies are slowly adopting agile. Agile methods are reducing the need for traditional models. The future of agile approaches seems very dominant. In general, there are some aspects of software project that can benefit from an agile approach and others can benefit from traditional approach. These results are shown graphically in Figure1.
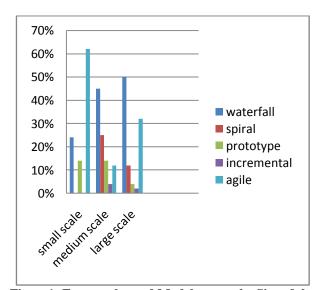

Figure1: Frequently used Model as per the Size of the Project.

3. Effects of Agile on Software Quality, Cost and Time:
As depicted in Table.2, when the issue of software quality was presented to the software respondents, 89% of the respondents were of the opinion that there was increase in the quality of software on using agile for small scale projects as compared to heavyweight.

TABLE II EFFECT OF AGILE MODEL ON SOFTWARE QUALITY

Values in %

Quality / Size	Quality Increases	Quality decreases	No Effect
Small Scale	89	5	6
Medium Scale	19	71	10
Large Scale	34	51	15

It was found that team members can communicate regularly throughout the development process while working on small projects. But most of the respondents believed that there was decrease in quality of the software on using agile for medium and large scale projects. These results are shown graphically in Figure2.
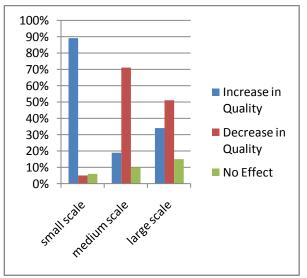

Figure2: Effect of Agile Model on Software Quality.

It is evident from Table3 that 54% software professionals were of the opinion that there decrease in the cost of the software on using agile for small scale projects. 41% believe that the cost will increase. Whereas, 5% believe that it would have no effect on cost. But when came to medium and large scale projects cost started to increase as compared to traditional models.

TABLEIII EFFECT OF AGILE MODEL ON SOFTWARE DEVELOPMENT COST

Values in %

Cost / Size	Increase in cost	Decrease in cost	No Effect
Small scale	41	54	5
Medium scale	54	41	2
Large scale	54	41	3

IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified

Vol. 3, Issue 11, November 2016

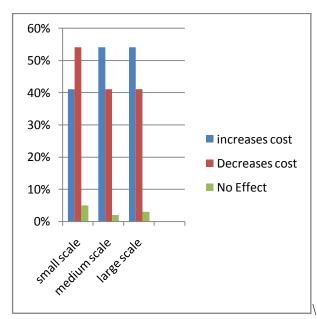These results are shown graphically in Figure3.

Figure3: Effect of Agile Model on Software Development Cost.

It is evident from Table4 that 17% of the respondents believe that there is increase in development time on using agile, 76% think that it will decreases the time and 7% were of the opinion that there would be no effect on using agile.

TABLEIV EFFECT OF AGILE ON TIME

Values in %

Time \ Size	Time Increases	Time decreases	No Effect
Small Scale	17	76	7
Medium scale	35	52	13
Large scale	61	32	7

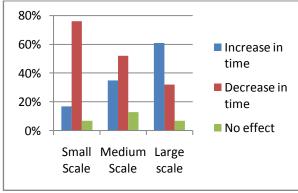These results are shown graphically in Figure4.

Figure4: Effect of Agile on Time.

It is clear from Table5 that 66% of the respondents were using spiral model for the development of risky projects because spiral model is the one that can identify risks early in the project.

Whereas, 17% favour incremental and 17% support agile as it minimizes risks.

TABLEV FREQUENTLY USED MODEL FOR THE DEVELOPMENT OF RISKY PROJECT.

Values in %

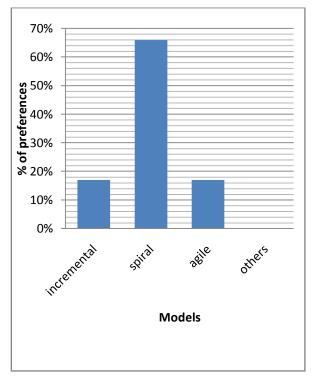| Models	Incremental	spiral	Agile
% of preferences	17	66	17

These results are shown graphically in Figure5.

Figure5: Frequently used Model for Risky Projects.

It is evident from Table6 that the most common problem experienced while using agile for small and medium size projects was lack of skilled a people [3].

Software professionals agreed that certain percentages of experienced people are required in an agile method.

Whereas, project complexity was found one of the major problem when using agile for the large size projects.

As discussed earlier, that agile proved to be good for small projects. It shows inverse relationship of agile with the size of the project.

TABLEVI PROBLEM IN PRACTISING AGILE METHODS

Values in %

Problem Size	Lack of communication	Lack of customer collaboration	Lack of top mgt. support	Lack of skilled People	Project complexity	Project team size
Small Scale	3	14	18	45	12	8
Medium Scale	4	18	25	29	16	8
Large Scale	7	16	11	18	34	14

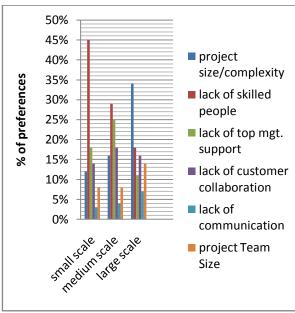These results are shown graphically in Figure6.

Figure6: Problem in Practising Agile Methods.

It is evident from Table7 that 62% software professionals support scrum for the development of small scale projects and 24% were using (XP) extreme programming.

Whereas, 69% software professionals favour scrum for the development of medium scale projects. In case of the development of large scale projects 72% software professionals adopted scrum and rest of the software professionals acknowledged other approaches.

TABLEVII FREQUENTLY USED AGILE APPROACHES FOR DIFFERENT SIZES OF SOFTWARE DEVELOPMENT

Values in %

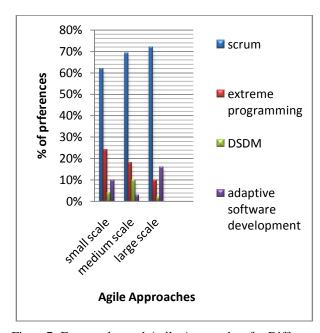| Agile Methods Size	XP	Scrum	DSDM	ASD
Small Scale	24	62	4	10
Medium Scale	18	69	10	3
Large Scale	10	72	2	16

These results are shown graphically in Figure7:

Figure7: Frequently used Agile Approaches for Different Sizes of Software Development.

It is clear from Table8 that 66% of the respondents were of the opinion that there is no process model that can fit in all situations.
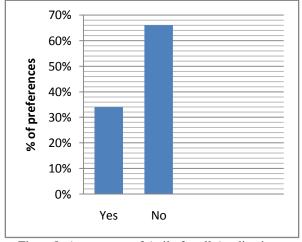

Figure8: Acceptance of Agile for all Applications.

One of the most important questions, why should we learn about these different methodologies instead of simply focusing on one? There is no one shoe that fits all [5]. Whereas, 34% agreed that agile can fit in all situation. These results are shown graphically in Figure8.

VIII. CONCLUSION

The analysis and discussion leads to the conclusion that agile model is frequently used by software developers independent of size of the project, as it has the flexibility to change the requirements at any stage. However, some of the software professionals preferred to use the heavyweight models for developing medium and large size projects.

Software professionals supported spiral model for critical projects where high risks are involved. It has also been found that Scrum and XP were used more frequently in organization as the most popular agile approach for the development of software's. Further, it can be said that the use of agile methods are increasing in comparison to heavyweight models. In general, there are some aspects of software projects that can be benefitted from an agile approach and others can benefit from traditional approach. Finally, it is concluded that there is no single process model that can fit in all situations. The selection of the models depends on the factors of the software project like size, cost and time.

REFERENCES

[1] Awad M.A.,"A Comparison between Agile and Traditional Software Development Models," The University of Western Australia 2005.

[2] David Cohen, Mikael lindvall, and Patricia Costa,"Fraunhofer center for experimental software engineering 4321 Hartwickrd, Suite 500 college park, MD 20742 USA".

[3] Dora Kumar Sujit and Dubey Pushkar,"software development life Cycle (SDLC) Analytical Comparison and Survey on Traditional And Agile Methodology", National Monthly Refereed Journal of Research in Science and Technology, Volume No. 2, Issue No. 8.

[4] Hawrysh and Ruprecht (2000). Lightweight methodologies: It's like Déjà vu all over again. Cutter It Journal.

[5] Khan Irshad Asif, Qurashi Jameel Rizwan and Khan Ali Usman, "A Comprehensive Study of Commonly Practiced Heavy and Lightweight Software Methodologies", Department of Computer Science Singhania University, Jhunjhunu, Rjasthan, India, vol. 8, Issue. 4, No.2, July 2011.

[6] Maheshwari Shikha and Jain Dinesh", A Comparative Analysis of Different types of Models in Software Development Life Cycle", International Journal of Advanced Research in Computer Science & software Engineering.

[7] Mishra apoorve and Dubey Deepty",A Comparative Study of Different software Development Life Cycle Models in Different Scenarios", international journal of Advanced research in Computer Science and Management Studies, ISSN: 2321-7782, Volume1, Issue 5, October 2013.

[8] Pekka Abrahamsson, OutiSalo, JussiRonkainen, Juhani & Wartsa," Agile Software Development Methods", Review and Analysis. Espoo 2002. VTT Publications 478. 107p.

[9] Qureshi M.R.J,"An improved XP software development process model", Institute of information technology Lahore Pakistan.

[10] Saeeda Heena, Ahmed Mukhtar, Khalid Hannan & Sameer Abu," Systematic Literature Review of Agile Scalability for Large Scale Projects", Department of Computer Software Engineering MCS, National University of Sciences and Technology, Rawalpindi, Pakistan, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 6, No. 9, 2015.

[11] Rupa Mahanti, M.S. Neogi and Vandana Bhattacherjee," Factors Affecting the Choice of Software Life Cycle Models in the Software Industry-An Empirical Study", Journal of Computer Science 8 (8): 1253-1262, ISSN 1549-3636.

[12] Venkat Subramaniam http://www.agiledeveloper.com/download. aspx accessed on 10 0ct 2016.

[13] First workshop on Agile Methods, Centre for Experimental Software Engineering Maryland. Accessed on 12 0ct 2016.