Impact of Development Methodology on Cost & Risk for Development Projects

Bishan Dayal Chauhan, Ajay Rana, Neeraj Kumar Sharma

Amity School of Engineering and Technology, Amity University Uttar Pradesh, Noida, India
HIMT, HIMT Group of Institutions, Greater Noida, UP, India
bishanchauhan@gmail.com , ajay_Rana@amity.edu

Abstract: **In software projects, first of the challenges in projects is to manage successful delivery of the software project with minimal cost and high quality of software by constantly reducing & mitigating delivery risk. Cost control, Risk reduction & mitigation and success of a project becomes key focus for project manager. For reduction of cost & risk we need to first have the bird's eye of view which factors influence the risk & and impact success of the project, having this birds eye of view gives a ray of hope if development methodology being adopted for software development can help to reduce cost, risk and consequently increase success rate. Let's consider a hypotheses: Software development methodology used for developing software impacts success rate of project, risk and reduce cost. This study analyses this laid down hypothesis and test this relationship for various projects. Let's group methodologies into two groups, for developing the software namely Classical Software development life cycle methodology group (Classical SDLC Methodologies) & Agile methodologies, each of these methodologies groups have multiple models under each of these. This study researched both of these groups to analyze impact on fulfilling our laid down objectives of reduction in cost & risk and increased success rate of the project by picking one methodology from each group, Classical SDLC is older methodologies group comprising of Waterfall, spiral, Incremental etc. whereas Agile is a newer methodology group, particularly, it is researched that if Agile methodologies is better fit than Classical SDLC methodology - Waterfall model for meeting our core objectives. We obtained results for various completed projects and evidence is analyzed for deriving the final conclusion.**

Keywords: **Software Development Life Cycle (SDLC), Waterfall, Cost Reduction, Project Success rate, Impact of Development Methodology, Risk Reduction, Agile Methodology**

I. INTRODUCTION

Imagine if you are driving to an important trip to a far-off place where you never went, no person will generally will go to such a journey without knowing name and the broader direction of the destination. Other critical considerations are the shortest routes and the distance. With this minimal information and with map person will at least feel more comfortable for completing the trip.

Software project management is more complex than planning a trip. So, there will be challenges in managing success, cost & quality of software. Choosing an appropriate software development methodology is just like taking a direction with its merits & demerits with respect to cost, risk and success of the project.

What Determines Success of a Software development project

Software development project success criteria's are

- Did the project completion is on time?
- Did the project completion is within its budget?
- How is the quality of software?
- Is scope is fully met?
- Is the software being developed is in good use.

In any software development project, scope, time & cost are three key factors. One cannot reduce the duration without changing cost or quality. So, to manage success of the project is to keep the cost & quality in check.

Software Development Methodologies (Pressman, 2001) [1]

"A software development methodology in software engineering is a framework that is used to structure, plan, and control the process of developing an information system - this includes the pre-definition of specific deliverables and artifacts that are created and completed by a project team to develop or maintain an application" (Pressman, 2001) [1]

Well researched methodologies are spiral development, prototyping, waterfall, incremental development and iterative, agile methodology and extreme programming and rapid application development,

"A wide variety of such frameworks have evolved over the years, each with its own recognized strengths and weaknesses. One software development methodology framework is not necessarily suitable for use by all projects. Each of the available methodology frameworks are best suited to specific kinds of projects, based on various technical, organizational, project and team considerations." (Pressman, 2001) [1]

There are several software development methodologies are used since the inception of information technology. Major of these are:

- SDLC methodology
- Agile methodology

Many models are there which fall into these methodologies:

- "Software development life cycle:
  - Waterfall: a linear framework
  - Spiral: a combined linear-iterative framework

978-1-5090-3012-5/17/$31.00 ©2017 IEEE

- Incremental: a combined linear-iterative framework or V Model
- Prototyping: an iterative framework
- Rapid application development (RAD): an iterative framework "(Pressman, 2001) [1]

- "Agile methodology:
  - Scrum
  - Extreme programming
  - Adaptive software development (ASD)
  - Dynamic system development method (DSDM)" (Pressman, 2001) [1]

Waterfall development (Pressman, 2001) [1]

"The waterfall model is a sequential development approach, in which development is seen as flowing steadily downwards (like a waterfall) through the phases of requirements analysis, design, implementation, testing (validation), integration, and maintenance. The first formal description of the method is often cited as an article published by Winston W. Royce in 1970 although Royce did not use the term "waterfall" in this article."

"The basic principles are:

- Project is divided into sequential phases, with some overlap and splash back acceptable between phases.
- Emphasis is on planning, time schedules, target dates, budgets and implementation of an entire system at one time.
- Tight control is maintained over the life of the project via extensive written documentation, formal reviews, and approval/signoff by the user and information technology management occurring at the end of most phases before beginning the next phase.

The Waterfall model is a traditional engineering approach applied to software engineering. It has been widely blamed for several large-scale government projects running over budget, over time and sometimes failing to deliver on requirements due to the Big Design Up Front approach. Except when contractually required, the Waterfall model has been largely superseded by more flexible and versatile methodologies developed specifically for software development. See Criticism of Waterfall model" (Pressman, 2001) [1]

Agile Development

Twelve Principles describe Agile (Beck, Kent, 2001) [2] [14]

i. "Our highest priority is to satisfy the customer through early and continuous delivery of valuable software"
ii. "Welcome changing requirements, even late in development. Agile processes harness change for the customer' competitive advantage"
iii. "Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale"
iv. "Business people and developers must work together daily throughout the project"
v. "Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done"
vi. "The most efficient and effective method of conveying information to and within a development team is face-to-face conversation"
vii. "Working software is the primary measure of progress."
viii. "Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely."
ix. "Continuous attention to technical excellence and good design enhances agility."
x. "Simplicity–the art of maximizing the amount of work not done–is essential."
xi. "The best architectures, requirements, and designs emerge from self-organizing teams."
xii. "At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly" (Beck, Kent, 2001) [2] [14]

Software Size & Effort

Sizing the software projects is a critical activity in software engineering which is used for estimating software size of the component / application for estimating software characteristics like defects, schedule, efforts, resources, etc. Size is as like a key characteristics of a software as weight is for tangible material. Size and effort are not same, it is important to differentiate between software size and effort. Size does not depend on technology but on the functionality and is an independent measure, effort depends on factors such productivity etc.

In general, we estimate size in kilo lines of code (KLOC), or function points, number of uses cases, count of objects. There are many methods for computing software size.

Example: If a software application of size 5000 lines of code or simply thirty function points is to be developed, and productivity (somewhat like speed) of engineering team is 1 Function Point / Day or 166 LOCs /day, then total effort required to develop the software can be computed as

Efforts Required = Software Size / Productivity

= 5000/166 = 30 Person Days

Or

= 30/1 = 30 Person days

II. OBJECTIVE

Objective of the Study

The objective is detailed analysis of methodologies used for software development particularly – Waterfall Methodology and Agile, to understand & analyze impact on cost, risk & success rate(s) of projects for software development and eventually facilitating strategic planning.

Often Project Managers or people responsible for success of project which includes on time & on budget completion have a big question in their mind that,

a) Which software development methodology should be used?
b) Will agile methodology help, to reduce risk & cost?

This study aims to understand if development methodology being adopted for software development can help to reduce cost, risk and consequently increase success rate.

Data Source

Data analysis being done using data from "Benchmarking Release 10 by the International Software Benchmarking Standard Group (ISBSG)" (ISBSG, 2011) [16]. "The ISBSG established in 1994 a not for profit organization that has been established to improve the global understanding of software practices, so that they can be better managed. ISBSG has gathered on 4,106 software projects from around the world, and made available on Release 10 of Estimating, Benchmarking & Research Suite CD."

S. No.	Project	Type	Methodology	Function Points	SDLC -Effort consumed (Person Days)
1	Project - 1	Case Study	SDLC-Waterfall	311	280
2	Project - 2	Case Study	SDLC-Waterfall	78	70
3	Project - 3	Case Study	SDLC-Waterfall	101	91
4	Project - 4	Case Study	SDLC-Waterfall	61	55
5	Project - 5	Case Study	SDLC-Waterfall	461	415

III. EXPERIMENT RESULTS

Agile Vs. SDLC: Efforts comparison in two Methodology

We have addressed questions being raised by experimenting on comparative analysis

a) Extensively worked on Methodology selection & agile methodology
b) Results leads to the analysis of the impact of methodology used.

Following steps were followed

i. Took the projects which are completed using Agile Methodology
ii. Efforts data is taken for these completed projects using Agile methodology as shown in Table 1: Software Projects Efforts using Agile Methodology for Completed Projects (Person Days)
iii. For sizing Function Point is used as a measure. Applying industry benchmark of productivity to estimate efforts for projects using SDLC-Waterfall methodology as shown Table 2: Software Projects Efforts using SDLC-Waterfall Methodology for Completed Projects (Person Days)
iv. Compared the efforts in two different methodologies as shown in Table 3: Comparison and Savings of Efforts for Agile vs. SDLC-waterfall Methodology

Table 2: Software Projects Efforts using Agile Methodology for Completed Projects (Person Days)

Source: ISBSG (2011) Data (generated using ISBSG data) [17]

S. No.	Project	Type	Methodology	Agile - Effort consumed (Person Days)	Function Points
1	Project -1	Case Study	Agile	195	311
2	Project -2	Case Study	Agile	54	78
3	Project -3	Case Study	Agile	58	101
4	Project -4	Case Study	Agile	40	61
5	Project -5	Case Study	Agile	298	461

To compute efforts, Productivity taken is as 0.9 FP / Person Day

Table 2: Software Projects Efforts using SDLC-Waterfall Methodology for Completed Projects (Person Days)

Source: ISBSG (2011) Data (generated using ISBSG data) [17]

Comparing efforts of development in SDLC-Waterfall approach & Agile Approach for five projects which are developed using Agile methodology, computing their function points and applying industry standard of productivity in the technology of projects and computing the effort in SDLC methodology, minimum effort saving % is coming 23% and maximum is 36% for agile methodology as shown in Table 3: Comparison and Savings of Efforts for Agile vs. SDLC-waterfall Methodology

Table 3: Comparison and Savings of Efforts for Agile vs. SDLC-waterfall Methodology

S. No.	Project	Efforts in Agile (Person Days)	Efforts in SDLC (Person Days)	Efforts Savings using Agile (Person Days)	% Efforts Savings using Agile
1	Project -1	195	280	85	30%
2	Project -2	54	70	16	23%
3	Project -3	58	91	33	36%
4	Project -4	40	55	15	27%
5	Project -5	298	415	117	28%

Also, as shown in *Fig. 1: Line Chart showing Comparison of efforts for Agile & SDLC-waterfall*
that agile efforts line is always lower than the waterfall efforts line, so less efforts are required in the agile methodology then in a waterfall methodology.

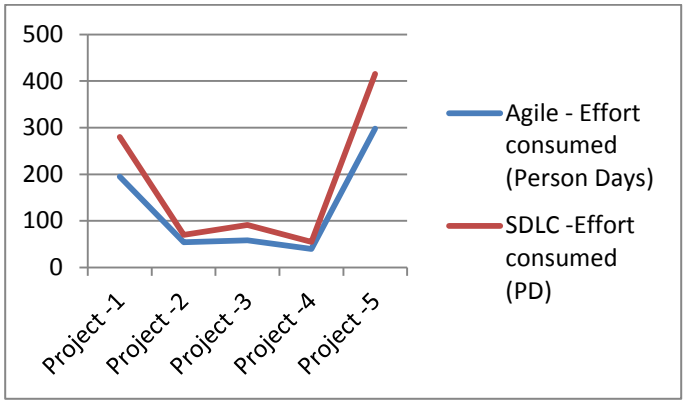

Fig. 1: Line Chart showing Comparison of efforts for Agile & SDLC-waterfall

Resource Optimization

"The agile approach to development is about agility of the Development process, development teams and their environment" (Boehm & Turner, 2004), [3].

"This approach incorporates shared ideals of various stakeholders, and a philosophy of regular providing the customers with product features in short time-frames (South well, 2002), [4]. This frequent and regular feature delivery is achieved by team based approach" (Coram & Bohner, 2005), [5].

"The development teams also have on-site customers with substantial domain knowledge to
help them better understand the requirements" (Abrahamsson, Solo, Ronkainen, & Warsta,
2002), [6].
Other observations coming from literature survey are

a. Optimum use of resources as sprints is done for each requirement delivery.

b. All resources in the sprint are well loaded and realization of contribution is more in all team members towards success.

c. Early use of software by end users so risk of end user acceptance is reduced / eliminated

d. This resource optimization leads to eliminating resource usage wastages

Requirement Prioritization – Most needed requirements are prioritized first

"Agile development methodologies emphasize rapid delivery of software products to the clients. According to (Boehm & Turner, 2005)[7], Fast cycles, frequent delivery: Scheduling many releases with short time spans between them forces implementation of only the highest priority functions, delivers value to the customer quickly, and speeds requirements emergence [7]."
"Agile methods are iterative and incremental development" [8], and "each successful completion of development iteration, it delivers software product increment to client, and thus agile software development is satisfying the customer through early and continuous delivery of the valuable software" (Garg, 2009) [9]. Whereas, "Traditional, lifecycle based software development delivers the software only after entire completion of development process and before that clients have no clear idea and view of software to be developed" (Moniruzzaman, Hossain) [10]
According to Pareto principle, it can be said that in software 20% of the features are used most and 80% features are rarely used or never, it becomes utmost critical to prioritize the requirements.
Few of the highly needed features can be taken first and some can be planned later. This way, it is possible to avoid few of the features which are envisaged earlier. So cost of these features is saved

IV. DISCUSSION

Software development methodology used for developing software impacts success rate of project, risk and reduce cost, in analyzing this laid down hypothesis, there are numerous factors which justifies the point –

i. There less efforts required to develop a software using agile methodology than the classical SDLC Method – waterfall methodology, i.e. efforts saved translates into cost savings or cost reduction.
ii. Prioritization is another technique used in agile method for developing the software and after delivery of this requirement & acceptance of this developed piece of software, new set of requirement is taken, so software acceptance risk by the customer is reduced.
iii. In agile method, a smaller scope is considered at a time, so always scope is controlled, and any changes scope are taken care in a subsequent cycle. Since Scope is small there is much higher chances of delivering on time

It is seen that agile method is influencing critical success factors like Scope, Cost, Delivery risk & time, so it impacts overall success of the project, our findings are also in line with Standish Group's CHAOS report -
"According to the Standish Group's, [11] famous CHAOS Report of 2000, 25% of all projects fail outright through eventual cancellation, with no useful software deployed. Sadly, this represents a big improvement over CHAOS reports from past years." [10]

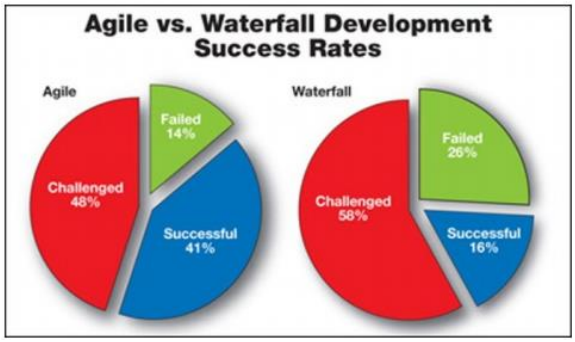

Fig 2: "Agile implementation success rate by The Standish group, (Source: http://blog.standishgroup.com/) [11]."[10]

These findings are also in line with the other published in "Agile development: Mainstream adoption has changed agility" - "in the past few years, Agile processes have not only gained increasing adoption levels; they have also rapidly joined the mainstream of development approaches" (West, Grant, Gerush, D'Silva, 2010) [12]

These findings are also well supported by work listed in "Agile software development: Impact on productivity and quality" – which states – "Traditional software development methods are not efficient enough to convene with the rapid change in requirements and short iterations that are required for efficient product delivery." (Ahmed, Ahmed, Ehsan, 2010) [13] Conducted "a quantitative analysis of the agile methodologies in practice, and the benefits associated with them."[13], In this analyses is "done on a defined set of attributes that result in increased productivity and improved quality through the application of these methodologies."[13]

Our findings are also in line with (Rico, D. F., 2008, 2004) [17][18] – "What is the ROI of agile vs. traditional methods" in which it conveys there is significant ROI (Return on Investment) using Agile methods.

However, there are few challenges needed to overcome, as specified by Boehm & Turner in – "Management challenges to implementing agile processes in traditional development organizations" – It conveys, managers "find that on small, stand-alone projects, agile practices are less burdensome and more in tune with the software industry's increasing needs for rapid development and coping with continuous change. Managers face several barriers, real and perceived, when they try to bring agile approaches into traditional organizations. They categorized the barriers either as problems only in terms of scope or scale, or as significant general issues needing resolution." (Boehm & Turner, 2005) [7]

To apply the agile methods in traditional environments, there are changes needed to be done as specified in – "The Organizational Changes Required and the Challenges Involved in Adopting Agile Methodologies in Traditional Software Development Organizations" (Misra, Kumar, Kumar, Grant, 2006) [15] which provides "a conceptual framework that would help managers for focusing on the important changes required and the challenges involved in agile software development projects."[15]

V. CONCLUSIONS

Often Project Managers or people responsible for success of project have a big question in their mind that,

a) Which software development methodology should be used?
b) Will agile methodology help, to reduce risk & cost?

These questions can now be answered – that agile methodology is better than SDLC-Waterfall methodology in terms of cost effectiveness of software development and there is lesser risk on development using agile method: So agile is a preferred methodology over conventional SDLC-waterfall methodology, and Software development methodology used for developing software impacts success rate of project, risk and reduce cost.

In this paper, we analyzed factors influencing success rate of the projects and analyzed these factors w'r't Agile & waterfall methods, as established in comparative analysis of efforts i.e. comparing efforts of development in SDLC-Waterfall approach & Agile Approach for five projects which are developed using Agile methodology, computing their function points and applying industry standard of productivity in the technology of projects and computed the efforts in SDLC-waterfall methodology, effort saving % is coming minimum 23% to maximum 36% for agile methodology.

Other observations coming from literature survey are:

All resources in the sprint are well loaded and realization of contribution is more in all team members towards success.

Early use of software by end users so risk of end user acceptance is reduced / eliminated

According to Pareto principle, it can be said that in software 20% of the features are used most and 80% features are rarely used or never, it becomes utmost critical to prioritize the requirements.

Few of the highly needed features can be taken first and some non-critical ones can be planned later. This way, it is possible to avoid few of the features which are envisaged earlier. So cost of these features is saved

So with these observations we conclude with 98% confidence that agile methodology is better cost effective than SDLC-waterfall mythology.

Our research gives factors which supports Agile Methodology as a better fit and establish it as an important methodology for cost reduction & increasing project success rate.

This is also in line with – "Agile methodologies have numerous advantages including that they: adapt very well to change and dynamism; are people-oriented and value-driven, rather than process-oriented and plan-driven; mitigate risks by demonstrating values and functionalities up front in the development process; provide a faster time to market; improve productivity (by reducing the amount of documentation) and will fail early/quickly and painlessly, if a project is not doable" (Abrahamsson, Solo, Ronkainen, & Warsta, 2002), [6].

REFERENCES

[1] Software Engineering – A Practitioner's Approach – Roger S. Pressman, 2001, Fifth Edition Published by McGraw-Hill

[2] Beck, Kent; et al. (2001). "Principles behind the Agile Manifesto". Agile Alliance. Archived from the original on 14 June 2010. Retrieved 6 June 2010.
[3] Boehm, B., & Turner, R. (2004, May). Balancing agility and discipline: Evaluating and integrating agile and plan-driven methods. In Software Engineering, 2004. ICSE 2004. Proceedings. 26th International Conference on(pp. 718-719). IEEE.
[4] Southwell, K. (2002). Agile process improvement. TickIT International Journal, 3-14
[5] Coram, M., & Bohner, S. (2005, April). The impact of agile methods on software project management. In Engineering of Computer-Based Systems, 2005. ECBS'05. 12th IEEE International Conference and Workshops on the (pp. 363-370). IEEE
[6] Abrahamsson, P., Solo, O., Ronkainen, J., & Warsta, J. Agile Software Development Methods. 2002. VTT technical Research Centre of Finland.
[7] Boehm, B., & Turner, R. (2005). Management challenges to implementing agile processes in traditional development organizations. Software, IEEE, 22(5), 30-39
[8] http://en.wikipedia.org/wiki/Agile_software_development
[9] Garg, A. (2009). Agile Software Development.
[10] Moniruzzaman, A., Hossain, S., Comparative Study on Agile software development methodologies
[11] http://blog.standishgroup.com/
[12] West, D., Grant, T., Gerush, M., & D'Silva, D. (2010). Agile development: Mainstream adoption has changed agility. Forrester Research
[13] A. Ahmed, S. Ahmad and N. Ehsan , "Agile software development: Impact on productivity and quality" , IEEE ICMIT , 2010
[14] "Agile Manifesto" Available: http://agilemanifesto.org/ principles.html
[15] Misra, S.C.; Kumar, U.; Kumar, V.; Grant, G. "The Organizational Changes Required and the Challenges Involved in Adopting Agile Methodologies in Traditional Software Development Organizations", *Digital Information Management, 2006 1st International Conference on,* On page(s): 25 - 28
[16] ISBSG "The Benchmark data for Software estimation" Release 10 (2011)
[17] Rico, D. F. (2008). What is the ROI of agile vs. traditional methods?
[18] Rico, D. F. (2004). ROI of software process improvement: Metrics for project managers and software