Journal of Information & Communication Technology
Vol. 10, No. 1, (Spring 2016) 105-120

AN OVERVIEW: MERITS OF AGILE PROJECT MANAGEMENT OVER TRADITIONAL PROJECT MANAGEMENT IN SOFTWARE DEVELOPMENT

Gul Ahmad[1]
Department of Management Sciences, SZABIST Dubai Campus
United Arab Emirates

Dr. Tariq Rahim Soomro[2]
Department of IT, SZABIST Dubai Campus
United Arab Emirates

Dr. Syed Mehmood Raza Naqvi[3]
Institute of Business & Technology (IBT)

ABSTRACT

This dissemination is a review study about the characteristics of both approaches agile (lightweight) & traditional (heavyweight) software project management methodologies and presented a comparison between them based on various aspects such as lifecycle, project success measurement, size of the projects, organization culture, management behavior, development team size, change tolerance; value added activities, planning & documentation. This research paper gathered data from various surveys conducted in last 10 years on factors of adoption trend & success rate of these two approaches which are shown in figure 10 & figure 11.

Key Words: Methodology; Agile; XP; Scrum; Heavy; Spiral; Waterfall; Traditional; Conventional; Plan Driven;

INSPEC Classification : A9555L, A9630, B5270

* The material presented by the author does not necessarily portray the viewpoint of the editors and the management of the Institute of Business & Technology (IBT)

[1] Gul Ahmad : gulahmadmcs@yahoo.com
[2] Dr. Tariq Rahim Soomro : tariqsoomro@gmail.com
[3] Dr. Syed Mehmood Raza Naqvi : dr.naqvi@gmail.com

© IBT-JICT is published by the Institute of Business and Technology (IBT).
Main Ibrahim Hydri Road, Korangi Creek, Karachi-75190, Pakistan.

1. INTRODUCTION

Software project management is the spotlight in recent digital era and is part from last more than five decades of the society. Initially the software programming were believe on code & fix sentiments and were developed with less upfront design plan. Code & fix concept was work for small size systems but with the growth of business information systems it head to more complexities and become harsh to fix issues without proper design plan. This plan driven concept ruled for many years on software development market but with emergence of recent changing business requirements an alternative approach was introduced in 2001 called as agile methodologies. The conventionalmodels like waterfall & spiralmodels are plan driven and eliciting & documenting complete set of customer requirements, architectural design and development detail plan also called as heavyweight. The agile in software development methodologies also known as light weight, agile manifesto mainly focus to emphasis on "people, communication & interaction, working software, customer collaboration, and welcome changes, rather than on processes, tools, contracts and plans"(B. Boehm.et.al,1998).

The core idea behind this research study to conducta detailed review study of both approaches traditional or heavyweight and agile or lightweight methodologies. For the plan driven approach, we included in my studythe two most popular methods Waterfall and Spiral models to discuss a high level review of their characteristics or attributes.Next section we shade light the vastly used agile methodologies Extreme Programming (XP) and Scrum models along with their holding characteristics. Furthermore, we carried out a comparative study of these approaches conventional & agile. In the next section we collected data from various most internationally known groupsincluding Tech Beacon, cPrime, StandishGroup, VersionOne, AmbySoft conducted surveys & reports on project success rate & adoption rate from 10 years. To conclude on the basis of gathered informations from various sources mentioned above which methodology is to usefor software development between agile & traditional approach

2. PLAN DRIVEN MODELS

Plan driven of heavyweight methods are considered to be the traditional way of managing software projects. These models are followed of step by step flow, such as feasibility study, gathering requirements, building the solution, validating &deployment. This methods believe on upfront documentation & complete design plan. There are many different heavyweight methodologies but thewell-known are Waterfall& Spiral models .

2.1 Waterfall

Waterfall model is a conventional approach consists of sequential phases such as requirements, design, implementation, testing or verification and maintenance with definite set of contents. Every phase of this model having related bunch of activities perform to prepare the output for the next phase of this model. These phases of traditional waterfall model are shown in figure 1 in detail. Each phase consists of related documents waterfall model are shown in figure 1 in detail. Each phase consists of related documents and pushing output to the next phase in sequence

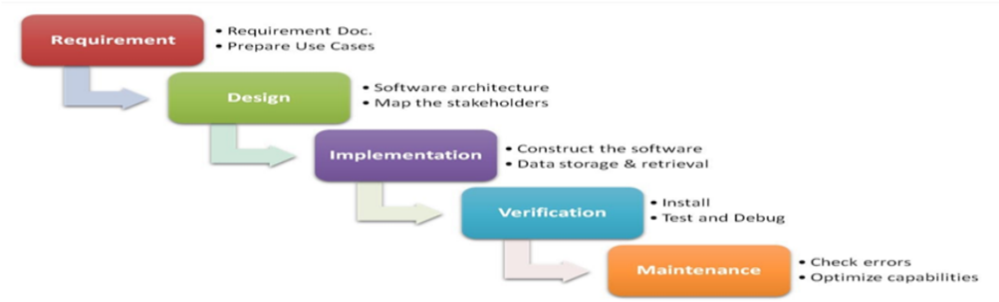


Figure. 1

2.2 Spiral Model

Spiral method is also a type of heavyweight software development approach, which trying to combines different components of designing & prototyping, in order to foster the merits holding by bottom-up and top-down constructs. In 1986 Barry Boehm described spiral model in his paper. This model has the following four main phases (J. H. a. A. Cockburn,2016).

- Requirements gathering – Solid aims & objectives are identified each cycle or phase
- Risk analysis and reducing prototyping – Major risks are defined, evaluated on impact basis and data is collected and analyzed to reduce the identified risks.
- Building & testing – A suitable approach followed for development & verification.
- Planning – The project developed via spiral model is reviewed and re-plan for next round of spiral cycle.

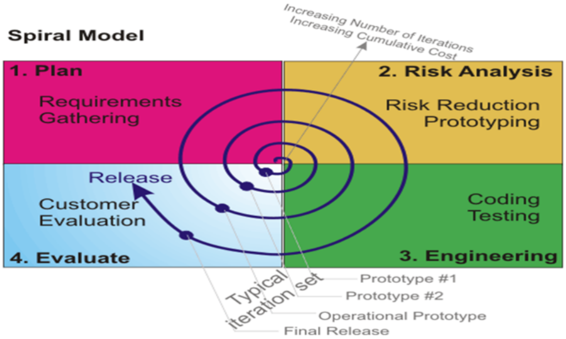


Figure. 2

2.3 Heavyweight Characteristics

Heavyweight or traditional models have been used for decades in project managements across the boundaries of nature discrimination.In the vintage this models were very popular in software market also for project management. Some common characteristics of heavyweight approaches are list below (M. Fowler.et.al,2016).

2.3.1 Predictive approach – Heavyweight or conventional methodologies believe on preplanned tendency along with detail documentation & layout design. In software development planning & documentation required a huge amount of time to details it.

2.3.2 Comprehensive Documentation – Complete customer requirements in the initiation of projects is the key factors in this approach of project management which is akaBDUF or big design upfront process. In this process gathering all of a user requirements prior to development.

2.3.3 Process& Tool Oriented–This approach is working on well-defined processes with sequential or parallel flow and also using some tools for project design & management.

3. AGILE MODELING

Agile methodologies or lightweight or modern models are is an alternative approach to conventional models used in software development market. These models well designed for frequently changing of requirements & unpredictability using sprints which are incremental, iterative work beats.In 2001 a group agile experts from various agile approaches decided to form aframework to better foster their sentiments which give birth to Agile Software Development. The core values ofagile manifesto are "We are uncovering better ways of developing software by doing it and helping others do it? Through this work we have come to value: Individuals and interactions over processes and tools working software over comprehensive documentation Customer collaboration over contract negotiation responding to change over following a plan". In the figure.3 shown the 12 principles of agile in the Agile Manifesto (G. A. T. R. S. M. N. BROHI,2014).

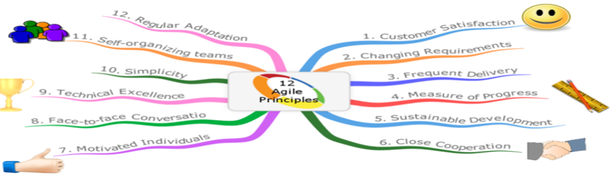


Figure. 3 (http://www.biggerplate.com)

3.1 Extreme Programming (XP)

Extreme programming or XP is very well known agile practice which consists by small cycles of development, incremental& iterative planning, continuous feedback, and propercommunication. XP team are proactive to respond to the changes of user requirements with courage. XP team members are working on coding, design, feedback, pair review & testing and refactoring.XP practices are shown below in Figure. 4.

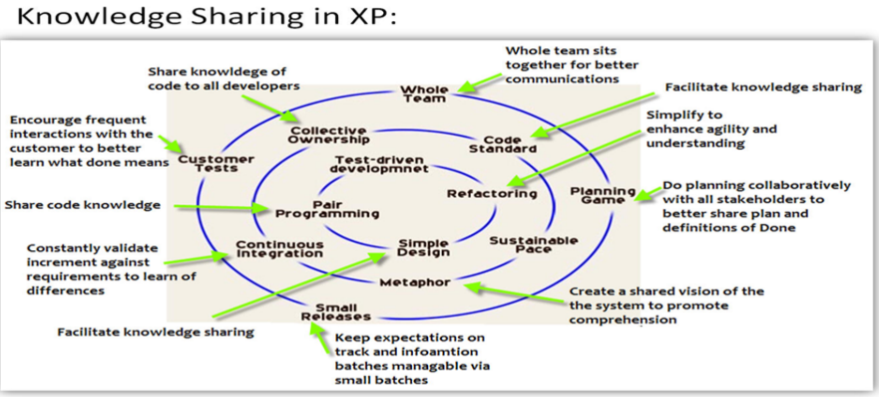

Knowledge Sharing in XP:
Whole team sits together for better communications
Share knowldege of code to all developers
Whole Team
Facilitate knowledge sharing
Encourage frequent interactions with the customer to better learn what done means
Collective Ownership
Code Standard
Simplify to enhance agility and understanding
Customer Tests
Test-driven developmnet
Refactoring
Planning Game
Do planning collaboratively with all stakeholders to better share plan and definitions of Done
Share code knowledge
Pair Programming
Constantly validate increment against requirements to learn of differences
Continuous Integration
Simple Design
Sustainable Pace
Metaphor
Create a shared vision of the the system to promote comprehension
Small Releases
Facilitate knowledge sharing
Keep expectations on track and infoamtion batches managable via small batches

Figure. 4

Extreme Programming project lifecycle are divided into 6 phases that are(a) Exploration, (b) Planning, (c) Iterations to release, (d) Production, (e) Maintenance and (f) Death. In first phase i.e. exploration the customer are responsible to writes user story cards, in planning phase prioritizeuser stories& schedule the first release. Testing & validation of the system can be done in the production phase, ideas & suggestions is done at the maintenance phase. Death Phase is the final product transition where no more stories are considered from customer.

3.2 Scrum

Scrum is an agile model having iterative, incremental process for development & management. Scrum main focus thatwhat the scrum team should follow or adopt to develop a more robust in the frequently anticipated of requirements change. Every scrum cycle or release produce a shippable working with potential functionality.Scrum practices flow are shown in Figure 5.

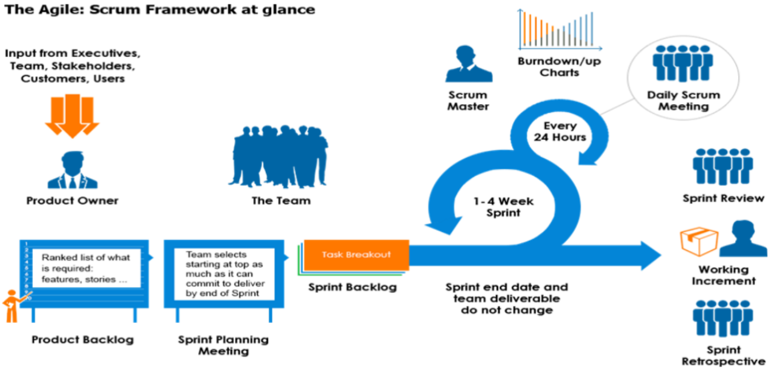

The Agile: Scrum Framework at glance
Input from Executives, Team, Stakeholders, Customers, Users
Product Owner
The Team
Scrum Master
Burndown/up Charts
Daily Scrum Meeting
Every 24 Hours
1-4 Week Sprint
Sprint Review
Working Increment
Sprint Retrospective
Ranked list of what is required: features, stories ...
Team selects starting at top as much as it can commit to deliver by end of Sprint
Task Breakout
Sprint Backlog
Sprint end date and team deliverable do not change
Product Backlog
Sprint Planning Meeting

Figure. 5

In Scrum process user stories are gathered as product backlog & prioritize them according to importance. Then some top level stories are picked from next release which is called sprint backlog. Each sprint has 1-4 weeks' time span, which produces a potentially shippable product released to customer. Each sprint has a daily scrum, sprint review & retrospective meetings.

3.4. Characteristics of Agile Methodologies

Agile models are a proactive reaction of frequently changing business models in emergence of technological advancements in business world which includes fragmentation, globalization & intangible products & services. Agile models have some common characteristics:

3.4.1. People Oriented

- According to agile manifesto "Individuals and interactions over processes and tools" people is most prestigious factors such as customers, agile team, stakeholders.

3.4.2. Adaptive – Embracing changes is the most visible fact of agile models, and agile team always welcoming changes at all stages of the project. In agile change factor foster learning curve to raise in corporate market to satisfy customer.

3.4.3.Balancing Flexibility & Planning

–Agile does not believe on predictive detail planning, in agile strategy is to plans for the next sprint, high level rough plans for the next month and keeping a cloudy clue about future of the product.Flexibility & revision is the core idea behind agile development, the work should be flexible that at any point of time can be revised or changed without difficulty.

3.4.4.Decentralized Approach

–In Agile development mostly decisions are making by the team which are may be disseminate around the globe. Agile managements are using to their analytical & decision making skills to clear the path for developers to move smooth on the project progress.

3.4.5.Simplicity

– Simplicity in agile world is the most important factor which means agile team always following the simple way to develop because for long on it's easy to make changes in the design if needed later. It means they are doing what they are asked not doing any extra for future use.

3.4.6.Collaboration

– Agile models are focusing on active involvement of customer & continuous feedback on often basis. Mostly customer are onsite with the agile team to work with them closely & openly without attempting any mistakes or missing of requirements. Agile also encouraging full time communication & collaboration among the members of agile team in all direction vertically & horizontally.

4. COMPARISON BETWEEN AGILE MODELS & TRADITIONAL MODELS

stream of project management and each of them have different characteristics. According to Boehm, the agile models main objective is on rapid delivery of value while on the other hand conventional approach is focus on high assurance. Heavyweight models behave that the requirements are completely defined & predictable to develop an extensive detailed plan while agile development focus on adaptive approach with high quality consist ofmulti skills small groups using the continuous improvement,rapid feedback & embrace changes. The following table emphasizes the primary comparison between agile methodologies & conventional methodologies .

Agile Project Management	**Traditional Project Management**
People-centric; Leadership and collaboration	Process-centric; Command and control
Focus on the short term scope regardless of detail plan & then moving to next release	This approach focus on developing of entire scope in start
Embracing changes because of flexible & adaptable structured procedures	Changes are always unwelcomed because of rigid structure of procedures
Agile team members have active communication & collaboration	Conventional team members environment are working individually with a less collaboration
Decisions are established as a result of active involvement & interaction in the systems	Decision & orders are made by hierarchical structures of organization
Agile team believe on continuous interaction, self-organizing &bare rules result in increased command over the system	Control promotion resulting in the increasing command & order
Agile adopted organizations should be flexible & base on necessary bureaucracy only	Conventional models adopted organizations are having rigid & static hierarchies & structures
In agile management roles are fostering & facilitators for the team	In traditional management believes on control
Employees are significant assets of the organization they all should contribute in system development regardless of roles & positions	Employees can be interchanged in organizations or can be replaced
Customer has active contribution & continuous onsite involvement throughout the lifecycle of the project	Customers are primarily contributing during the collection of requirements & transitions stages
Iterative& incremental approaches to prioritize user stories with continuous feedback stakeholders tends to results in valuable production shipment in a short period	The complete project scope are breakdown in small chunks arrange in a hierarchical structure called work breakdown structure (WBS) and assigned resources
Future control on projects are limited due to uncertain, unpredictable scope & risks. Therefore agile models have limited & short plan detail.	Projects scope & certainties of risks are predictable and can be documented in advance planning with details.
In agile models documentation are very simple & limited, can be done merely when	Conventional models based on detail documentation for the system in advance &

they needed	planned thoroughly
Agile team believe on the principle of self-directed feeling free to release deliverables followed by the agreed schedules	Project manager are the director of the team working in traditional environment & they tightly bound to deliver the product on planned schedules
Requirements are gathered as user stories throughout the project lifecycle. In agile models the final product could be different from the outset projected one.	Requirements defined in initial phase of project lifecycle. In this model scope creeping is always anticipated due to customer ask for changes.
Developers are performing testing of their code & refactoring it continuously. Team receiving constant feedback from customer. Team members are learning from their past mishaps & jargons.	User testing &feedback on the product from customer are take place at the end of project. Because of late involvement & feedback from customer can emerges huge problems & could be expensive to fix after complete product is developed.
Its team responsibility to assess project scope & direction constantly. Team are self-directed & have the authorities to change product direction to meet the customer needs to keep them satisfied & sustain in the long run.	Teams in traditional environment work on project that can span years to deliver the final product which can leads to no relevancy because due long time the business needs have been changed.
The agile development life cycle is Iterative, incremental &adaptive which also informally called evolutionary-delivery model	Traditional development life cycle is linear, structured, predictive &anticipated. It is also called predictive cycle model
This approach is more flexible& adapted for better understanding of collectively contextual needs to offer rapid faster development	This approach is universal approach providing solution, predictability & high assurance

Table. 1

4.1 Major Agile benefits over Traditional approach

Every approach has merits & demerits in their our domain of development & management such as conventional models have significant success in construction, oil & gas projects while agile methodologies are famous for the success rate in software market. The following are some important aspects highlighted for agile having significant benefits over traditional approaches.

4.1.1. Rapid delivery of software products

Agile or lightweight methodologies emphasize the principle of rapid& fast transition of products to customerby small releases.Short sprints or cycles, frequent

delivery& scheduled sprints of 1 to 4 weeks of time span contains of high prioritized user stories which foster the rapid delivery of product to customer. Agile methods are having
iterative and incremental development structured models and each iteration delivering a potential shippable product which increment the final product in different releases. Because of rapid delivery & customer onsite involvement keep the customer satisfied & sustained.

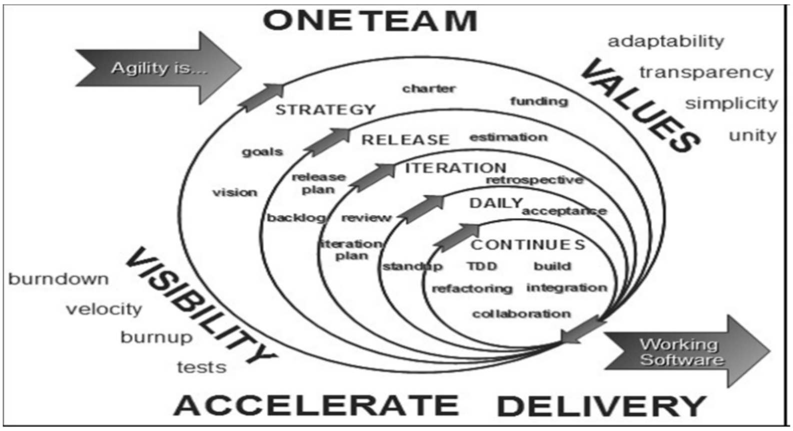


Figure 6: Iterative process and incremental delivery software products [4].

4.1.2. Highly tolerant of change requirements

Agile approach embracing change in user requirements at any stage while conventional models are feeling reluctant in this phase. To make change in the traditional methods are complex & rigid to implement which may cause to failure of projects while embracing changes is the agile most significant principle. Agile has the ability to respond the customer changes which leads to learn & success of project in the market. One of the factor of agile emergence was the frequent asking for changes from customers which are not only difficult in Heavyweight methods but some time looking impossible.

4.1.3. Reduce cost and time

Traditional approach relative cost of changes in requirements or cost of fixing errors are very high, either they were missed or lack of understood, throughout the lifecycle of product development. Agile development involves less cost of development as rework, management, documentation and other non-development work related cost is reduced.It is clear that approach thatthe team should test frequently & in outset stages. Agile also foster to reduce the feedback loop, consumption of time between development & validating it which cause to short the time & cost tremendously of product lifecycle.

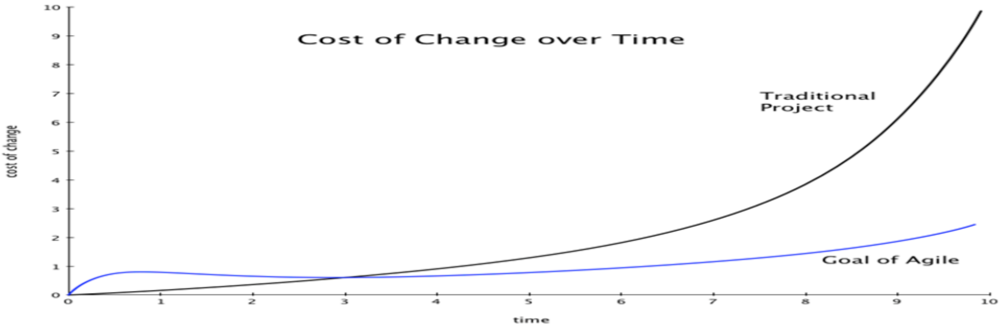


Figure. 7

4.1.4.Early feedback from Customer

In conventional models are following predictive & upfront plan approach which is informally called BDUF (Big Design Up Front) and BRUF (Big Requirements Up Front) development tactics. In traditional methodologies details plan, design & requirement documentsare developed in the initiation. While agile models on the other hand working on small releases, user stories with no or less documentation & short term plan which leads of customer feedback at the early stages of product development & continuously are involved in the lifecycle.

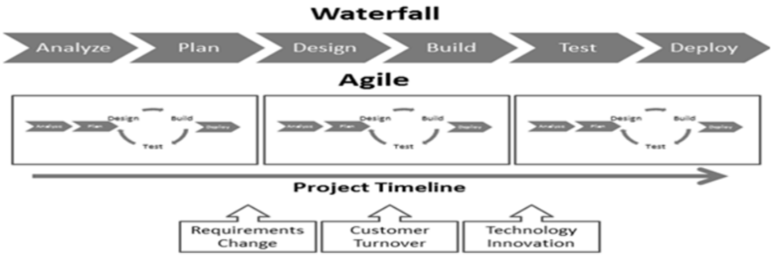


Figure 8: design phase composition between waterfall and agile development.

4.1.5. Documentation

Agile development models are focusing on value adding activities and reducing activities which are non-value added activities like comprehensive documentations, detail planning & future values of the product. Agile more emphasis delivering the current product nor more or less.

4.1.6.Focus on high quality product

In lightweight methodologies emphasize on rapid delivery, short term release, higher customer satisfaction, low defect rate and a proactive response to requirements changes. Plan-driven models are focusing on predictability, high assurance & stability.In agile models the onsite customer involvement with development team assuring the product quality, customer satisfaction & product success.

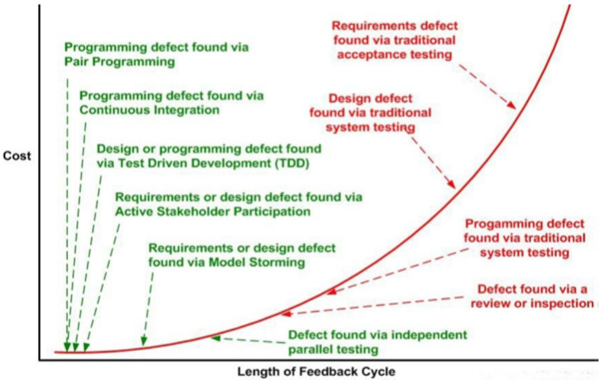
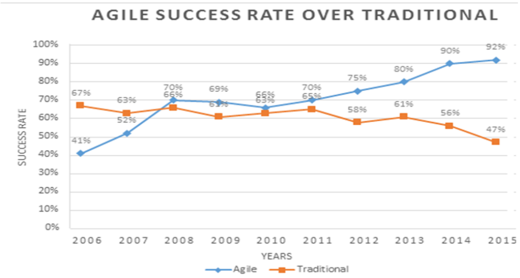


Figure 9: Comparison of Feedback cycles of agile vs traditional approaches. (http://www.ambysoft.com)

5. AGILE PROJECTS SUCCESS RATE

Traditional methodologies are mostly challenges in delivery of software projectstherefore most of the software projects are failed or late & over budget. Agile approach main principle to reduce waste, eliminate non-value added activities and over-production in order to deliver the product what customer actually needed. In Agile models are focusing on small releases & rapid development & delivery with short term sprints & onsite customers involvement, getting continuous feedback from clients are all driving forces to successful delivery of the product on time & within the budget. InFigure.10 the data are gathered from different surveys conducted by well-known organizations that are showing the success rate of agile projects over traditional projects of the past 10 years.



6. ADOPTION OF AGILE

In today modern software market different industries are seeing a greater return on investment (ROI) in IT projects. From financial institutions to corporate size organizations are facing on large scale changes in technology platforms, payment gateways, financial, assets and risk management systems, while attempting to deliver products & services as the customers demand or need. From online banking, e-payments trends toward increasing in mobility worlds & m-payments with digital revolution which needs to rapid response, tracking of accounts & economics to reduce financial risks. In Figure 11 data are gathered from different survey conducted by world known organization from 10 years which showing the adoption of agile development models over conventional models.

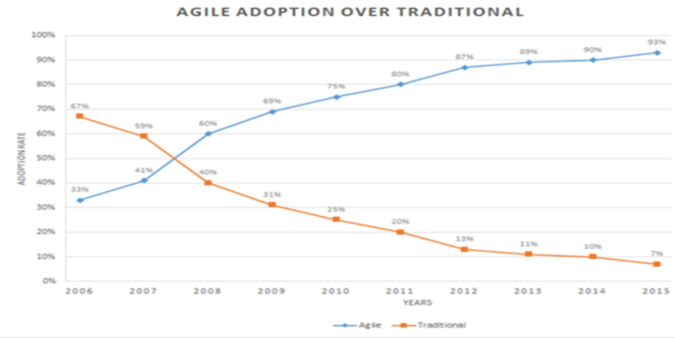


Figure 11: Agile development degree of adoption.
(, , , , http://www.ambysoft.com)

CONCLUSION

Global competition is at an all-time high. Technology is advancing at an unprecedented pace. Organizations must deliver more with fewer resources. Although there is no perfect solution to project management and success, executives and managers are turning to agile project management as a key solution to assist in this challenge. In today dynamic market & increasing trend to deliver quality products in the information & business era with rapidly changes in global market lets the professionals to think a different way of developing software which emerges agile models. Predictive project approaches are tightly regarded in project management & were using for decades which were successfully implemented in different organizations. However, for today dynamic & global complex IT market, traditional models can be stay inefficient & ineffective as the business requirements are not tangible. Agile models have emerged due highly iterative and incremental nature of process, where the development team & stakeholders are work together to identify the business needs & mutual understanding of the domain. All stakeholders are onsite involved in defining the product requirements, what needs to be built& deliver, and prioritize functionality by demand of customer.

Agile software development models are iterative and incremental nature approaches

&become popular in software information industry. According to the data collected in the survey many organizations moving to adopt agile regardless of size from small to large level organizations. In this dissertation, we conduct the study of main benefits of agile in the contrast of traditional approach which tremendously improves software development processes. We also provide with this research report, the success rate and current adoption state of agile software development with help of data collected from different surveys results. This study provided an in-depth benefits of agile models in the software development industries over conventional approach and also providing details side by side comparison of these two development approaches that Agile Models & Traditional Models.

REFERENCES

B. Boehm, "A Spiral Model of Software Development and Enhancement," in IEEE Computer, May 1998 .

M. Fowler, "The New Methodology," http://www.martinfowler.com/articles/newMethodology.html, 29 2 2016.

K. Beck, "Embracing change with Extreme Programming," in IEEE Computer, Vol. 32, 10 October 1999.

W. Cunningham, "Agile Manifesto," 29 2 2016.

K. Beck, Extreme Programming explained: Embrace change. Reading, Mass, Addison-Wesley, 2004.

L. A. Williams, "The XP Programmer: The Few-Minutes Programmer," IEEE Software, 2003 .

L. R. a. N. S. Janoff, "The Scrum software development process for small teams," IEEE Software, no. 17, pp. 26-32, 2000 .

K. a. K.Schwaber, "Experiences of using Scrum with XP," control chaos, 2016.

J. H. a. A. Cockburn, "Agile Software Development: The Business of Innovation," IEEE Computer, 29 2 2016.

B. Boehm, "Get ready for agile methods," with care. Computer, vol. 35, no. 1, pp. 64-69, 2002.

T. &. D. T. Dyba, "What do we know about agile software development?.," Software, IEEE, vol. 26, no. 5, pp. 6-9, 2009.

S. M. R. &. M. G. Nerur, "Challenges of migrating to agile methodologies.," Communications of the ACM, vol. 48, no. 5, pp. 72-78, 2005.

D. S. A. H. A B M Moniruzzaman, "Comparative Study on Agile software development methodologies," IJTM/IJCEE, 2013.

Consulting Firms.," Umeå School of Business, 2009.

P. Wendorff, An Essential Distinction of Agile Software Development Processes Based on Systems Thinking in Software Engineering Management, Addison Wesley.

J. Stapleton, Dynamic systems development method – The method in practice, Addison Wesley, 1997.

E. Scanlon-Thomas, "Breaking the Addiction to Process: An Introduction to Agile Project Management," IT Governance Publishing , 2011.

F. E. A. &. M. F. Paetsch, "Requirements engineering and agile software development. In Enabling Technologies: Infrastructure for Collaborative Enterprises," in WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on IEEE., 2003.

D. Marks, "Development Methodologies Compared," in N CYCLES software solutions, N CYCLES, December 2002.

J. D. Luca, "Feature Driven Development Overview," 29 2 2016.

C. &. B. V. R. Larman, "Iterative and incremental developments. a brief history," Computer, vol. 36, no. 6, pp. 47-56, 2003.

C. Larman, Agile & Iterative Development: A Manager's Guide, Addison-Wesley, 2004.

P. Kruchten, "What is Rational Unified Process?," The Rational Edge, 2016.

J. Highsmith, Agile Software Development Ecosystem, Addison Wesley, 2002.

J. A. Highsmith, Adaptive Software Development: A Collaborative Approach to Managing Complex Systems, Addison Wesley, 2000.

D. &. R. G. Greer, "Software release planning: an evolutionary and iterative approach.," Information and Software Technology, vol. 46, no. 4, pp. 243-253, 2004.

M. Fowler, "Is Design Dead?," Software Development, 29 2 216.

S. P. a. J. Felsing, A Practical Guide to Feature-Driven Development, Upper Saddle River, NJ, Prentice-Hall, 2002.

T. &. D. T. Dybå, "Empirical studies of agile software development: A systematic review," Information and software technology, vol. 50, no. 9, pp. 833-859, 2008.

A. Dagnino, "An evolutionary lifecycle model with Agile practices for software development at ABB. In Engineering of Complex Computer Systems," in Proceedings. Eighth IEEE International Conference on (pp. 215-223). IEEE., 2002.

L. W. a. A. Cockburn, "Agile Software Development: It's about Feedback and Change," IEEE Computer, pp. 39-43, June 2003.

B. &. T. R. Boehm, "Management challenges to implementing agile processes in traditional development organizations," Software, IEEE, vol. 22, no. 5, pp. 30-39, 2005.

K. S. a. M. Beedle, Agile Software Development with Scrum, Upper Saddle River, NJ, Prentice – Hall 1st Edition, Oct 2001 .

C. R. J. N. B. Grady, "Object Oriented Analysis and Design with Applications," 2nd edition, Addison Wesley Longman, 1998.

S. W. Ambler, "Duking it out," Software Development, 2002 .

B. M. V. B. A. C. A. C. W. F. M. &. T. D. Beck K., "Manifesto for agile software development," in The Agile Alliance, 2002-04, 2001.

A. S. Koch, "Agile software development: evaluating the methods for your organization," Artech house, 2005.

K. &. W. C. Petersen, "A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case," Journal of Systems and Software, vol. 82, no. 9, pp. 1479-1490, 2009.

B. &. Z. E. A. Bahli, "The role of knowledge creation in adopting extreme programming model: an empirical study," Enabling Technologies for the New Knowledge Society: ITI 3rd International Conference IEEE, pp. 75-87, 2005.

C. J., "Issues and Challenges of Agile Software Development with Scrum," Issues in Information Systems, vol. 9, no. 2, pp. 188-195, 2008.

D. J. Reifer, "How good are agile methods?.," Software, IEEE, vol. 19, no. 4, pp. 16-18, 2002.

R. K. Wysocki, Effective project management: traditional, agile, extreme, Wiley., 2011.

B. &. T. R. Boehm, "Using risk to balance agile and plan-driven methods," Computer, vol. 36, no. 6, pp. 57-66, 2003.

J. Charvat, "Project management methodologies. Selecting, implementig, and supporting," 2003.

M. Awad, "A Comparison between agile and traditional software development methodologies," School of Computer Science and software Engineering, The University of Western Australia, 2005.

M. M. B. G.-M. Marian STOICA, "Software Development: Agile vs. Traditional," Bucharest University of Economic Studies Romania, Informatica Economicã , vol. 17, p. 4, 2013.

T. R. S. M. N. B. Gul Ahmad, "XSR: Novel Hybrid Software Development Model (Integrating XP, Scrum & RUP)," International Journal of Soft Computing and Engineering (IJSCE), vol. 2, no. 3, pp. 2231-2307, 2014.

Direction," EUROPEAN ACADEMIC RESEARCH , vol. 1, no. 11, pp. 2286-4822 , 2014.