A Study of Hybrid: Validation of Factors Affecting The Adoption Of Agile-Waterfall Hybrid Methods In Software Development

Ishank Jain
Department of Computer Science
University of Waterloo
Waterloo, Canada
ishank.jain@uwaterloo.ca

***Abstract*—Many organizations small or large are transitioning to hybrid software development approach as part of their continuous push towards better flexibility and shorter lead times on a software project, yet some companies still continue to stick to traditional waterfall and agile software development methods. Hybrid "agile-waterfall" models have become increasingly prominent, mostly building one development model by combining elements of the two methods. Blending the two methods requires better understanding of these methods and may lead to problems; if one wants to adopt a hybrid approach, they first need to know the factors (A factor can be a practice, methodology or tool that support or oppose an approach in a study) that can influence its adoption. In this project, I conduct a qualitative research, and I review available literature to identify the challenges an organization might face and how they mitigate them on their way to adopt "Agifall" for iterative and rapid software development. The paper summarizes advantages and disadvantages of both the waterfall and agile methods in order to understand and blend key characteristics that combine best practices to meet an organizations' needs. [20] [10]. Further, I review case studies conducted at organizations from different corporate spheres to learn the needs for a successful hybrid model [10] [25] [23] [12] [2] [26] [18]. From the case studies I identify several factors that affect the adoption of a hybrid method and deliberate how each factor can impact its adoption by an organization. Finally, I was able to conduct a semi-structured interview with the System development team at a MedTech company in Canada to try to validate the results of the findings. Increasing our understanding about these hybrid approaches will aid in improving existing practices and developing new efficient methods.**

***Index Terms*—Agile development method (ADM), Hybrid methods, Large-Scale development, Software Development Life Cycle (SDLC), Waterfall development methods (WDM), Water-scrum hybrid.**

I. Introduction

Software Development Life Cycle is an organized process of specifying, designing, deploying, documenting, and supporting software. The traditional 'Waterfall' development method is a sequential linear approach to development that has distinct project life cycle phases (ref Fig 1); here the previous phases must be completed and verified before moving on to the next one [7]. The waterfall approach is well-defined, gated with upfront requirements specification, and assumes the project timeline is predictable. The need for the plan driven processes included in the waterfall method, originate from heavy manufacturing industries such as aerospace, where robustness and correctness are usually important concerns [21]. The waterfall method provides recognized useful characteristics, such as well-documented functional & technical requirements which can aid efficient knowledge transfer when team members are not collocated. According to (Spundak, 2014) [28], the Waterfall method tends to give best results when teams are distributed and get new team members frequently, but these methodologies are often considered too rigorous, inflexible, and a bit old-fashioned for many software development projects [19].

Alternatively, the 'Agile' development method is characterised by iterative, exploratory, evolutionary and incremental processes, where developers and stakeholders actively work together to identify what needs to be built, and prioritize functionality [7] [24]. For instance, agile methodologies promote a close collaboration with the stakeholder, self-organizing teams, and responsiveness to change over inflexible processes [16]. There are many ADM approaches such as Kanban, Scrum, Extreme Programming (XP) that can be implemented as suited to organizations' development needs. In ADM, a large amount of work is divided into smaller repetitive steps called 'iteration' plan (called 'sprint' in scrum); an iteration is developed and tested in a parallel fashion (ref Fig. 2). Useful characteristics of the ADM include stakeholder involvement throughout the project to provide feedback and refine the product and its features; iterative testing of each iteration plan while it is being developed; allowing the development team to be creative and innovative; fewer time is spent to prepare extensive documentation; and following an adaptive planning method with plenty of room to accommodate change requests as they come [28].

The Agile approach is significantly different from the Waterfall approach in its characteristics and implementation. Transitioning between methods is not easy for any organization; Even though ADM provides many advantages, its adoption does not come without any challenges for the organization [6] [17]. Moreover, agile might not be the perfect fit for all type of projects [14], for instance, in many ADM projects scaling agile may bring challenges, including coordination between several agile teams, lack of up-front requirements which defines the direction of project, lack of pre-emptive requirements analysis (cost, tools, man power, resources etc.), as well as all the challenges of distributed projects, as for many globalized organizations [11] [18].

To address the issues some organizations have developed hybrid project development methods which combines charactersticks from both waterfall and agile methods to address the challenges that they face with the current practices [5]. Lack of understanding of the hybrid model may lead to organizations to not appreciate what they are getting in adopting hybrid model, which could limit potential benefits and possibly bias benefit assessments [12]. ADM can benefit from WDM by providing clear guidance on how to initiate and deliver the project, communication management, managing project integration, cost management, as well as risk management. While, WDM can benefit from ADM's self-reliant teams, flexibility in delivery, and acceptance of continuous adjustment; ADM keeps stakeholder involvement throughout the project to get feedback and reduces documentation. Both waterfall and agile methods have advantages and disadvantages, so it is not possible to one-sidedly assert that one method is better than the other [28] [21]. Through hybrid approach, the intention is to promote Agile practices in a Waterfall method [1]. Generally, the Waterfall model is used to define the upfront work, i.e. to plan the functionality, features needed to be developed for the project, the budget, or other organizational aspects, however, the model still allows flexibility to incorporate changes. Afterwards, agile is used to implement the software development process with regular upgrade, testing, and stakeholder updates. Finally, the processes such as the delivery, are again carried out using the Waterfall methodology [25].

The main objective of this paper is to add to the current understanding of the hybrid development method by analyzing the hybrid approach through literature and case-studies. The approaches described in the case studies [10] [25] [23] [12] [2] [26] [18] combines the elements of the Agile method into Waterfall which provide sufficient information for analysis and to understand hybrid method in real world environment. In this paper, I first demonstrate the key characteristics of waterfall, agile and hybrid development methods. Second, it presents the summary from the review of the case-studies conducted at major international organizations. Finally, I conducted a qualitative study in which I examined the use of hybrid development practice at a Medtech company in Canada; in the study I try to address the research questions described in the next section.

Section III describes the research methodology used for academic review, conducting questionnaire survey, interview and its plan. In section IV, the paper describes the theoretical basis of the waterfall, agile and hybrid development methods. In section 4, we analyze the factors identified from the review of literature and case-studies. Further, I validate the findings through a qualitative study conducted at the MedTech company w.r.t. to the identified factors. The final section of the paper concludes by summarizing the findings and presenting future research directions.

II. Motivation and Rationale of the study

I got the inspiration for this studies for the reason that proof/research is required for a software development practice to gather acknowledgment in the industry. This paper is important to uncover the advantages of hybrid development method and reveal usefulness they offer over traditional practices. Thus, there is a need to supply the readers with an outline on this topic and I expect that this paper fills that hole.

The value of any study can be understood from how well it elaborates on previous work, also from the study's intrinsic properties. Thus, combining all the unbiased, credible results from previous research and verifying it through qualitative study would be a step towards understanding and expanding our knowledge on the domain. Similarly, the rationale of my study was to present credible results and benefits from the overwhelming amount of publications through a critical exploration and evaluation of the previous studies as well as present the results from the qualitative study to validate the factors for adoption of hybrid model.

To bridge the gap from what we know about, and to what we may not yet understand about hybrid development approach. Development is at the heart of every organization, therefore, there is a need to understand the characteristics, advantages-disadvantages of hybrid development method and how adoption of hybrid development cycle is affected at an organization.

III. Research Questions

The goal of this paper is to study and contribute to the literature by identifying the factors that affect the adoption of hybrid development method in an organization. To achieve this, I will address the following research question in this paper:

1) What are the best practices from ADM and WDM that can be combined together to successfully plan, manage, and deliver projects at large organizations?
2) Which factors determine whether an organization should adopt hybrid agile-waterfall method over traditional methods?
3) How do these factors influence the adoption of hybrid development approach?

IV. Research Methodology

The goal of this paper is to uncover factors that support or oppose the adoption of hybrid development method. To follow up on this, I conducted the literature study and qualitative study which consists of six different stages:

- Looking for the literature – collecting data,
- Assessing the search results – selection of academic papers for primary studies which included comparison of

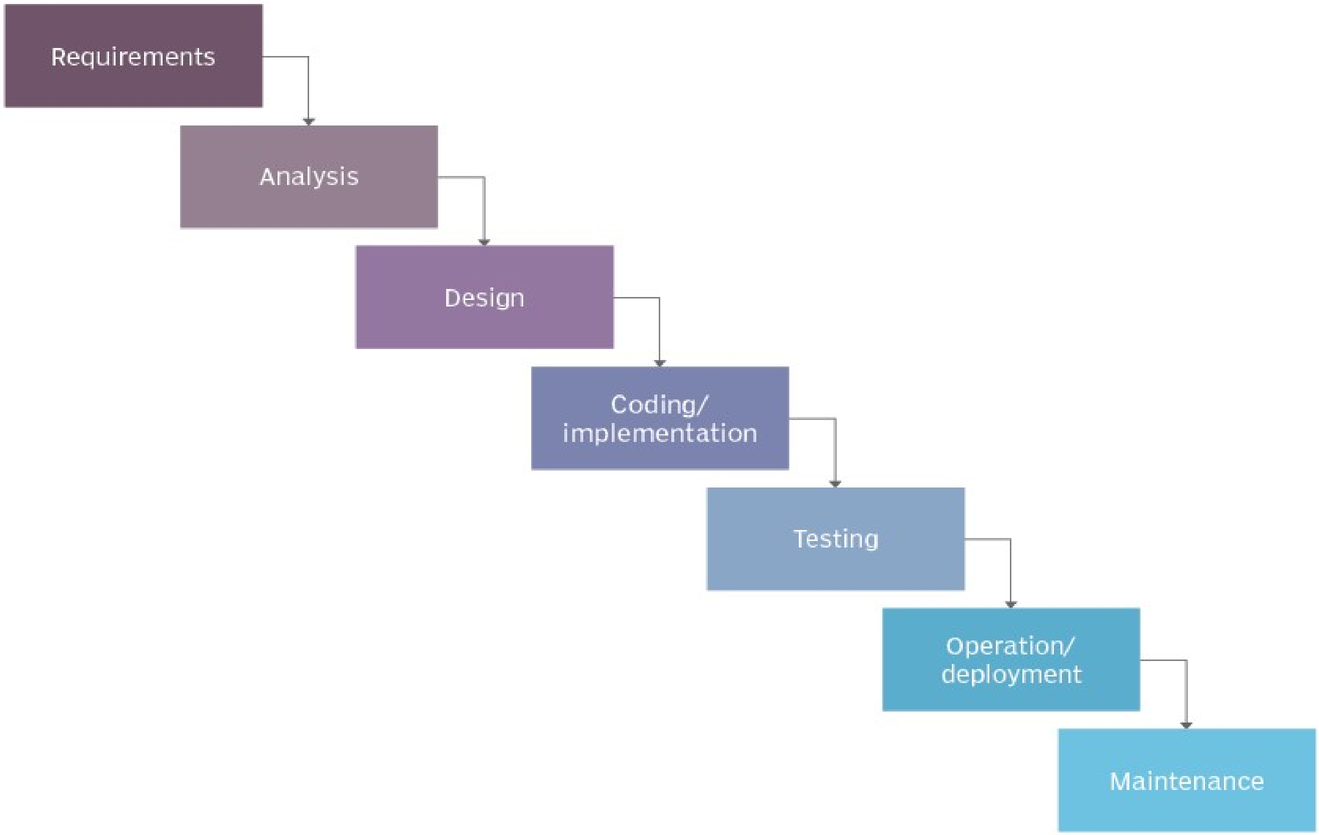

Fig. 1. Waterfall Development Model (source: https://www.educba.com/waterfall-model/)

the development models as well as case studies conducted at various organizations,

- Reviewing the papers,
- Synthesizing the results,
- Conduct and analyze questionnaire survey and semi-structured interview to validate the findings,
- Reporting the results.

The style of qualitative research method has been adopted from (Schlauderer et. al. 2015) (Kusters et. al. 2017) with slight adjustments to achieve project goals. This method allows me to gain in-depth knowledge on the hybrid model and identify factors. In the first stage, my goal was to gather the appropriate academic papers, articles and journals for my study. For that purpose, I extensively and iteratively searched popular databases for academic resources, this includes IEEE Software, IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), SpringerLink, Harvard Business Reviews, International Conference on Enterprise Information Systems (ICEIS), Communications of the ACM, and ACM Digital Library. The search terms included hybrid development methods, agile development, waterfall development, software development models, issues with traditional model, ground for adopting hybrid development approach etc. The time frame of the search for the review is bound within the last two decades from 2000-2018, in which emergence and adoption of hybrid development method has grown rapidly.

In second part, I reviewed the academic research and case-studies to identify the factors that impact the adoption of hybrid method. The factors are selected if they were described in more than 1 case-study (ref. Table 1) and how relevant they are to the hybrid model. This paper summarizes key highlights from each case-study and lists all the factors in Table 1. In the discussion I try to verify the identified factors using the statements from the interviewee and questionnaire survey conducted at a Canadian Medtech SME. The purpose is to be able to substantiate the claim made in the case-studies. To keep the company name and interviewee name anonymous: the company is simply addressed as 'Canadian Medtech Company' in the paper, and Interviewees are called called Interviewee 1, 2, and so on.

The qualitative study consists of two parts: a) Questionnaire survey, and b) Semi-structured interview. The questionnaire survey consisted of 11 questions and was necessary to evaluate the interviewee response without any bias from the interviewer. The questions in the survey were randomized to avoid any structured order, for instance, questions related to 'flexibility' offered by hybrid approach in terms of team productivity

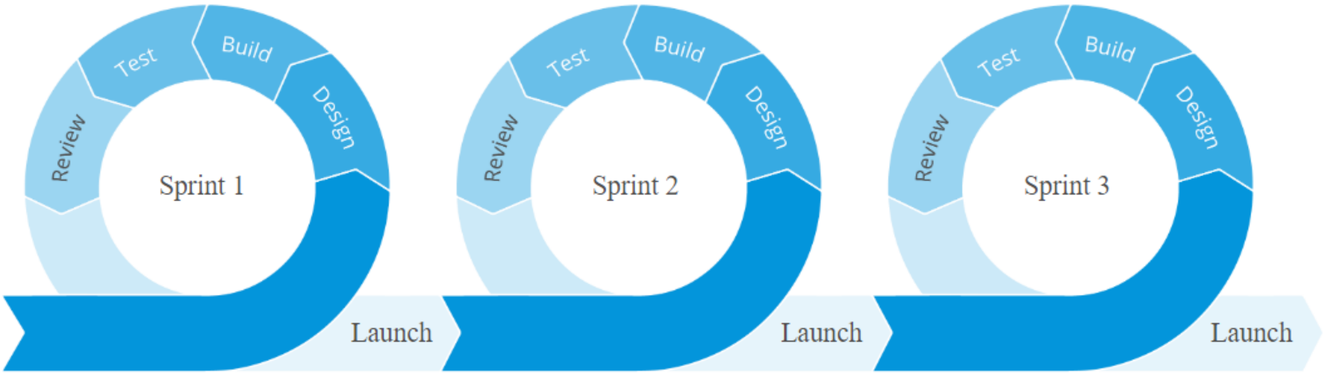

Fig. 2. Agile Development Model (source: https://www.mendix.com/blog/agile-process-why-you-need-feedback-loops-both-during-and-after-sprints/)

or product deployment were not asked one after the other in the questionnaire. After analyzing the responses from the questionnaire I conducted a semi-structured interview to understand the reasoning behind responses from the interviewees. The questions were directed towards software development practices at the organization and how the team adapts their practices with the change in needs of the customer.

The scope of the project is limited to the papers published between time period 2000-2018 due to time constraints. The interpretation from analysis of the hybrid model is limited to corporate sectors included in the case studies in the academic review. At last, the project is also limited in extent of survey and interviews due to time constraint and man hours.

V. Literature and Related Work

In this section, the paper describes what is waterfall and agile software development, their important characteristics, and what are their advantages and disadvantages. Second, we discuss why organizations should be interested in hybrid approach, as well as challenges and success factors.

SDLC process begins from conception to completion of a software project i.e. the software development team is responsible for the complete project development life cycle. After doing initial software requirements specification (SRS) and requirements analysis; developers follow different models of SDLC used in the organization.

A. *Waterfall Development Method*

The Waterfall approach for design and development was the first established modern approach to building a software system. This method was originally defined by Winston W. Royce in 1970. It quickly gained support from the development team because everything flows logically from the beginning of a project through the realization of software, and was easily applicable [8]. The Waterfall method begins with the assumption that all requirements can be gathered up front during the Requirements phase [9]. Communication with the stakeholder is front-loaded into this phase, as the Project Manager/ Team Leader does best to get a detailed understanding of the stakeholder's requirements. Once this stage is complete, the process runs "downhill" [29]. The steps included are Requirements Analysis, Design, Implementation, Testing, Verification, and Maintenance (ref Fig1)

Waterfall method emphasises on the sequential movement from one phase to next, each phase must be finished in order to start the next phase. Through the end of each phase a documentation is developed which enables efficient knowledge transfer and supports narrative for decision making [15] [2]. An independent quality assurance(QA) team produces the test cases to evaluate whether the software fully satisfies the requirements outlined in SRS. Different versions of waterfall change the Requirements phase into the Idea phase [8], or split the Requirements phase into Planning and Analysis [29]. Some other models further break the Design phase out into Logical and Physical Design sub-phases [29]. However, the basic underlying principles remain the same.

The Waterfall method have certain advantages, which includes: a) Well-documented requirements and functional specification, b) Extensive documentation in waterfall model can aid efficient knowledge transfer when team members are distributed in different locations. Conversely, the Waterfall method carries with it quite a few disadvantages, which includes: a) possibility of changing requirements during the development cycle, b) project can often take substantially longer to deliver, c) A bug identified at later stage can cost significantly more to get fixed or result in project being scrapped [4].

B. *Agile Development Method*

One of the main complaints from the developers with the waterfall method is the lack of communication with the stakeholder beyond the requirements phase. This results in inability to respond to changing stakeholder requirements [13]. To counter this limitation ADM uses an iterative process where all the project managers and developers actively collaborate with stakeholder who provides feedback throughout the entire process of developing software. In ADM, the work for the developer team is divided into smaller iterative chunks called

Iteration Plans (also called sprints in Scrum agile method). This implies that in each iteration- the product is designed, developed, tested, and reviewed in a parallel fashion (ref Fig 2). After completion of each iteration a small part of the software is launched rather than the whole product. This means that testing is not a separate phase but an integral part of the development process. The main aim of this is to ensure early identification of bugs and issues which can effectively control project cost [4]. Teams in Agile are self-organized and have the flexibility of choosing a task to work on.

The development in Agile is lead by a project manager who guides small teams, and there responsibilities include encouraging teamwork, elaborating the roles and responsibilities of the members, establishing communication between stakeholder and team, and providing the needed recourse to accomplish the task [3]. The Agile method have certain advantages, which includes: a) Change requests are easier to implement , b) Agile does not spend time on extensive documentation, c) The stakeholder are involved throughout the project, and provide give feedback to tweak the product. Conversely, the Agile method carries with it quite a few disadvantages, which includes: a) Teamwork is highly essential, if team members do not cooperate well, the project will face failure, b) Transferring knowledge which are gained from one-to-one meeting with stakeholder is hard to transfer to new members, c) Merging development by individual member can be challenging and may induce time-delay, specially, if the teams are distributed.

Many researchers believe Agile is better than Waterfall and some believe that change is needed. However, both approaches have significant differences, as demonstrated through published work [27] [2]. Some academic also explained approaches that consolidate both WDM and ADM [5] [7].

C. Hybrid Development Model

The blended software development method is the idea of combining the waterfall and agile methodology characteristics to manage projects more effectively [21] (ref Fig 3). As learnt from the academic review, majority hybrid approaches have the following characteristics:

1) Requirements are defined, i.e. a vision is formulated and the project is initiated [2] (Waterfall).
2) It is able to adapt to changing requirements, i.e. the stakeholder has the possibility to evaluate the software after each iteration to add or refine requirements (Agile),
3) Should be Frequently testable (Agile),
4) Ability to communicate what the project will deliver (Waterfall),
5) Time management, labour management and cost controlling techniques, (Waterfall),
6) Adjustable meeting hours and channels to communicate i.e. allowing flexible routines for knowledge transfer and information sharing between members & teams (Agile).
7) Documented - which makes it Simple to understand and easy to transfer knowledge to new members, though the amount of documentation can be less extensive (Waterfall-Agile).
8) Works well for both small as well as large projects.

Several sources provide examples of hybrid methodologies. Binder et. al. (2014) [5] "present the blended development model starting with an initial high-level plan, followed by detailed planning at each iteration, leading to the final implementation. Paper [22] describes hybrid approach called "Waterfall-up-front" to specify requirements and reduce confusion about project objectives and deliverables, followed by an agile method in the design, implementation, and unit testing phases to speed up the process and reduce rework, and completing the project with "Waterfall-at-end" for high-level testing and acceptance" [21]. The idea here is to suggest characteristics that can be blended from waterfall and agile to form a hybrid technique that supports the organization in best possible way. In next section the paper presents summaries from review of case-studies conducted at organizations operating in diverse sectors.

VI. Analysis from case-study

In this section the paper highlights requisites of hybrid development approach, probable issues and major findings from case-studies reviewed for academic literature. In the subsequent section, the paper summarizes the most noteworthy factors that influence and characterize the adoption of hybrid method.

A. Agile with Discipline

A team of researcher from De Paul University conducted a study at IBM, Chicago to examine a hybrid Agile approach that combines WDM-ADM practice called 'Agile with Discipline' [2]. This study examines how Agile is integrated with traditional development practices at IBM. The Agile-with-discipline integrates agile development components into a structured approach to project development.

The key highlights from this case study include:

1) The Study revealed that the project managers are heavily engaged in "leading people, unlocking resources, and supervising the project to success".
2) The requirements are defined up-front in this model; "we Start any project we at least try to get a blueprint or an initial set of requirements" [2]. It is required to establish project plan even though the budget and scope are loosely defined.
3) The study shows that this hybrid approach has sufficient documentation and flexible timelines to accommodate requirement changes, sprint development, and continuous customer feedback.
4) The members of the team may interchange role if necessary. For Instance, in certain scenario a senior developer or stakeholder may temporarily take role of project manager.
5) Documentation is essential part of this hybrid approach to ensure knowledge transfer and record requirement

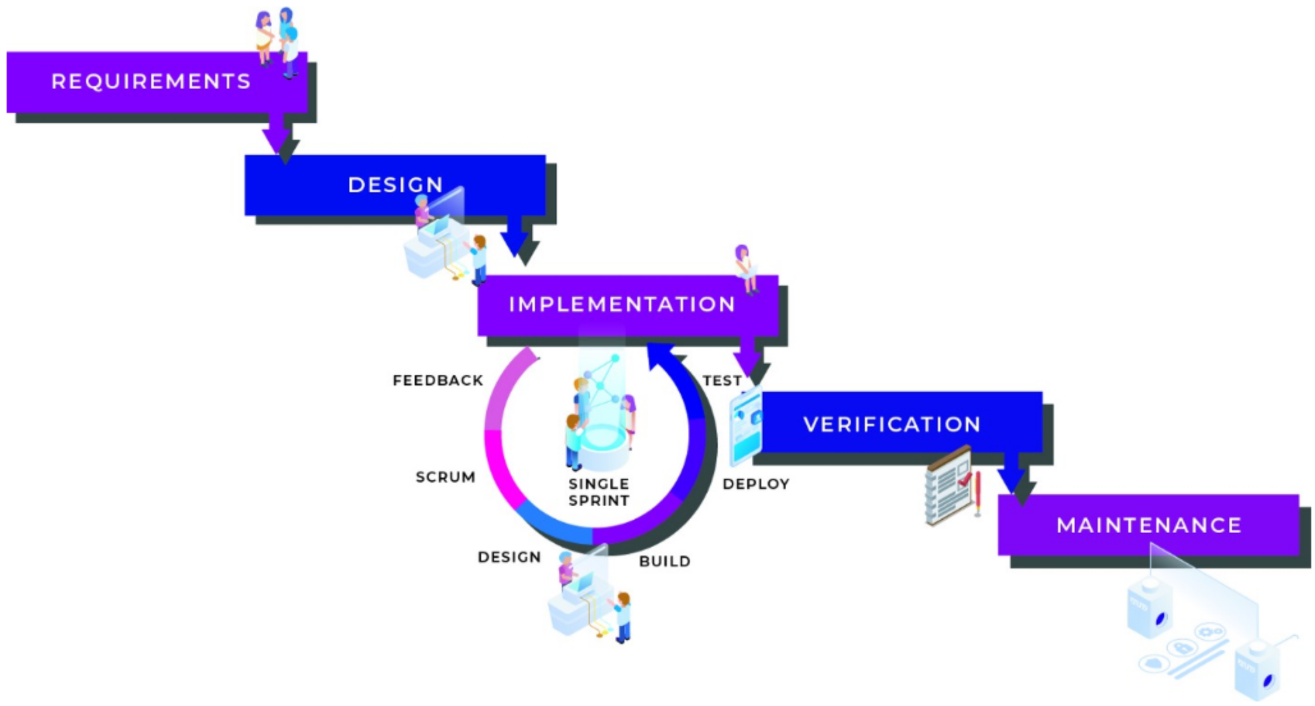

Fig. 3. Hybrid Development Model (source: https://medium.com/2359media/effective-agile-waterfall-hybrid-project-management-25af260aaa6d)

changes. “any changes to the timeline or the budget is reflected based on new request changes” [2]. Not all team members at IBM are co-located. Thus, Documentation and collaboration are important to keep the members engaged and move the project forward.

B. *Hybrid method at a Dutch financial Institution*

This case-study was conducted at a financial organization in Netherlands. The aim of this study was to “map issues (risks and problems) at the interface of agile and waterfall development method which impacts coordination and cooperation.” [10]. The issues were classified into 22 categories from 28 total identified from literature survey and were later verified via case-study.

The key issues that were identified from literature survey in this paper include:

1) Overlap in tasks and responsibilities creates confusion and duplication of work.
2) Absence of design documentation before the start of the development process can result in lack of communication and cooperation across development teams. Additionally, this might result in a final product which can be different from expectations.
3) Lack of active exchange of knowledge prevents new insights from arising.
4) Lack of proper client representation in stakeholder development team.
5) Delay in implementing changes by central control team results in extended deadlines. This adds to backlog and increased project cost.

The case-study was able to validate all 22 identified issues, but was not able to add to the list of issues via interviews. Once the issues are identified these can be potential factors that affect the perception of hybrid model in an organization. This paper has categorically guided the pattern of my qualitative research. The case-study appropriately reflects substantial issues that an organization faces when trying to adopt a new development methodology.

C. *Qualitative study conducted at a Swiss bank*

In this study, the organization uses Water-Scrum-Fall method, which combines Scrum with waterfall method to implement a hybrid approach. The paper discusses several acceptance factors and investigate how they are perceived based on the Diffusion of Innovations theory (Rogers 1995). The DOI Theory assumes “that the individual adopters have a different willingness to assimilate an innovation and that, consequently, the users will assimilate an innovation at different times” [25]. The study addresses the following research questions: “Which factors determine the acceptance of hybrid agile-traditional methodologies?; Compared to traditional development methodologies, are these factors perceived as drawbacks or benefits?.” The case-study evaluates several factors from the perspective of DOI theory to answer the above research question.

The factors studied in this case study which result in the adoption water-scrum-fall approach are studied through categories in DOI theory, which include (ref Fig (4):

1) Relative advantage
2) Compatibility

3) Complexity
4) Trialability
5) Observability

"Relative advantage, compatibility, trialability, and observability positively affect the rate of acceptance, whereas complexity is negatively correlated" [25]. The case-study reveals that requirements are better defined in water-scrum-fall (hybrid) method which provides a dedicated phase to clarify development goals and to facilitate the upfront planning. In addition, the study reveals that the stakeholders are satisfied with there involvement through continuous daily meetings. However, self-responsibility is below the desired level that impacts collaboration, communication, flexibility and transparency among team members. Besides, the study reveals that the process complexity is lower then traditional practices, however, extra efforts are required to integrate each developers' work.

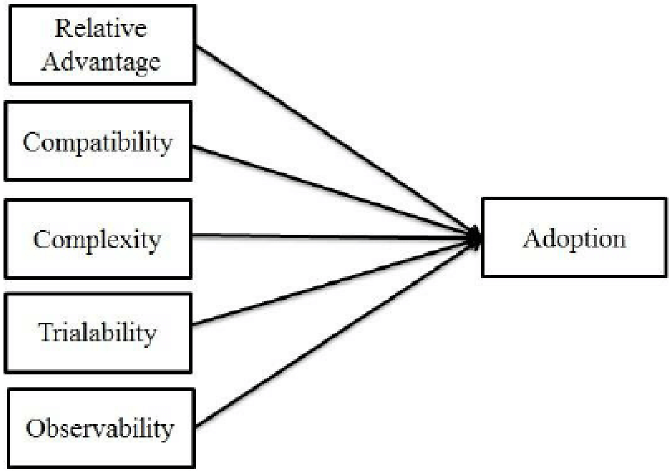

Fig. 4. Diffusion of Innovation Theory (Roger, 1983)

D. Software Development at Microsoft

The study was conducted at Microsoft in 2005 to study Human Interactions in Programming, developers' typical tools and work habits and their level of satisfaction at Microsoft [23]. The most notable take-away is that developers create and maintain mental models of code instead of documenting the code. Since, the developers are keeping only mental notes, they require frequent meetings with individuals to keep the development in sync. This causes interruptions and "greatly slow the rate at which a newcomer to a team can become productive." and makes it harder for the team lead to observe what are the developments in the project.

The key highlights from this case study include:

1) In this hybrid approach, the developers tend to keep "mental models" of there code which makes it harder to Understand the rationale behind a piece of code.
2) Frequent interruptions caused by regular meetings (agile method) has been identified as problem.
3) This approach lacks documentation, thus, transferring knowledge to a newcomer is described as a challenge. The paper suggests that a newcomer requires a mentor to help them start with the work.
4) The approach at microsoft uses 'Code Modularity' which results in code duplication and additional time is needed to understand and debug the code.

E. Large-Scale Agile Transformation at Ericsson

The objective of the study conducted at Ericsson between 2013-2014 was to observe how the organization experimented with different set-ups for their agile teams aiming for rapid end-to-end development. The research goal was to investigate how Ericsson supported adoption of agile into waterfall at its distributed centers. In this case-study, authors conducted semi-structured interviews in and uncovered many key points [18].

The key highlights from this case study include:

1) In this case-study, lack of a defined process (up-front requirement) was not considered a major problem. Instead, it provided flexibility to the team to adapt changes as the project moves forward.
2) Issues were observed with the component based structure: it was challenging to plan and coordinate work, thus working across teams was defined as "chaotic". The project development was modular and caused problem at time of integration.
3) The method adopted at Ericsson required experts and consultants to help with knowledge transfer, thus "component teams and a sequential, waterfall type, process." was used at multiple sites.
4) The lack of documentation and addition of new members connected via virtual means hindered the development process.

F. Hybrid in Manufacturing sector

This case-study presents an empirical research to compare agile and waterfall methods for new project development. The case-study proposes a hybrid framework to address the challenges arising from the differences in managing the two methods (Waterfall Agile). This case-study has been included for providing a perspective from the manufacturing sector [26].

The paper breaks up the framework into 5 categories, which include:

1) Project architecture: the study revels that defining product scope allows easier propagation of local changes to the product. Moreover, unexpected extra work resulting from resolving change requests requires extra hands and thus cross-team roles are beneficial. A team leader plays important part for knowledge transfer in cross-team roles.
2) Team roles and structure: the study recommends hiring specialists are to understand stakeholder's requirements and explain it to the team. This is necessary step if the organization is building in-house software and needs additional expertise on the final product.
3) Project Plan: For upfront specifications, the study recommends to define 'gates' (milestones); this is required

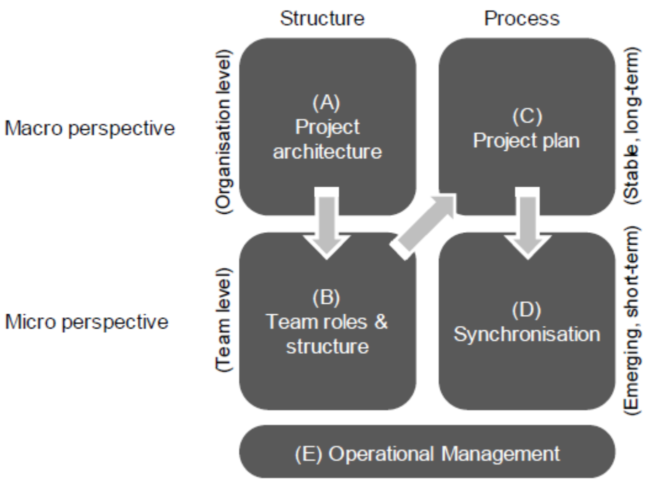

Fig. 5. Proposed Hybrid structure (source: Schuh et. al., 2017)

to close knowledge gaps and make key decisions, so that the goal is clear.

4) Synchronisation: the study identified lack of inter-team exchanges, thus,cross-team roles such as part-time domain experts and architects are proposed in the case-study.
5) Operational Management: the study recommends for hybrid agile team to adopt "discipline, process obedience and precise task definitions through minimal but essential documentation such as "cost forecasts" and "risk assessment" [26].

G. Control Changes in a Hybrid Development Environment

"Control relationships play an important role in helping organizations achieve their objectives." [12]. This paper describes the change in control relationship within an organization when adopting agile methodology into waterfall method. The paper explores following research question: "How does the introduction of the Agile method in a traditional Waterfall-oriented software-development environment change the controller-controlee relationship in projects and across functional boundaries?". The case-study is conducted at a fortune 100 company to validate and observe there successful implementation of the hybrid approach.

The key highlights from this case study include:

1) The iterations help the developer team to dynamically pace their work to attain anticipated end product characteristics that stakeholder desired.
2) "Sprint iterations" replaced the time allocated for planning, developing, and testing in the Waterfall software-development process.
3) The documentation is needed for requirement lock-down during iteration planning, and feedback at regular intervals provides a mechanism for the end user to understand the requirements.
4) The "iteration demo activity" provided an opportunity for product managers to monitor their team members relative to other contributors in the team.
5) In hybrid control, product managers use the "iteration demo and iteration retrospect activities of the Agile methodology to provide feedback" [12]. Feedback is documented and incorporated in future developments.
6) Developers have less control over requirement prioritization and the ability to determine which requirements are completed first.

VII. Results and Discussion

In this section, paper summarizes a list of factors that emerged as most noteworthy from the review of case-studies described above. I cite statements from the interviews I conducted at Medtech company to try to validate the factors observed in the hybrid development method. The data gathered shows that the interviewees working in the System development team working at a Canadian Medtech Company had profound experience with the Waterfall model as well as agile methods. Each person interviewed had an average of 6.2 years of professional experience in software development. Using the statements from the interview, I tried to support or oppose specific factors for the hybrid method, on which the interviewees reached broad consensus. Below is the table with all the identified factors and case-studies that describe them in their hybrid model.

TABLE I
Factors Identified From Case studies.

Identified Factors (F)	Case-Study
Requirement Specification	A, C, F, G
Communication and Collaborations	A, B, C, D, F
Individual responsibility & Motivation	C, D, E, G
Documentation- Knowledge Transfer	A, B, D, F, G
Cross-team roles	A, F
Flexibility and Project completion	A, B, F
Complexity	C
Values and Work Culture	B

In table 1, alphabets represent the sub-section in section 6 under which the case-study was summarized. In my study I took six factors that were described in more than one case-study.

In discussion of each factor I mention whether the characteristic is adopted from either waterfall or agile development practice.

A. F1: Requirement Specification

Q1:How important is it for you and the developer team to have requirements specified before the work starts on a product? Do you make changes to specifications during the project?

"Upfront requirements are the first step before starting any coding. This is kind of contract between the business / process team with the team." "As [developer] team, we would not start any development until we have minimum requirements and commitments from the business/process team. Without clear

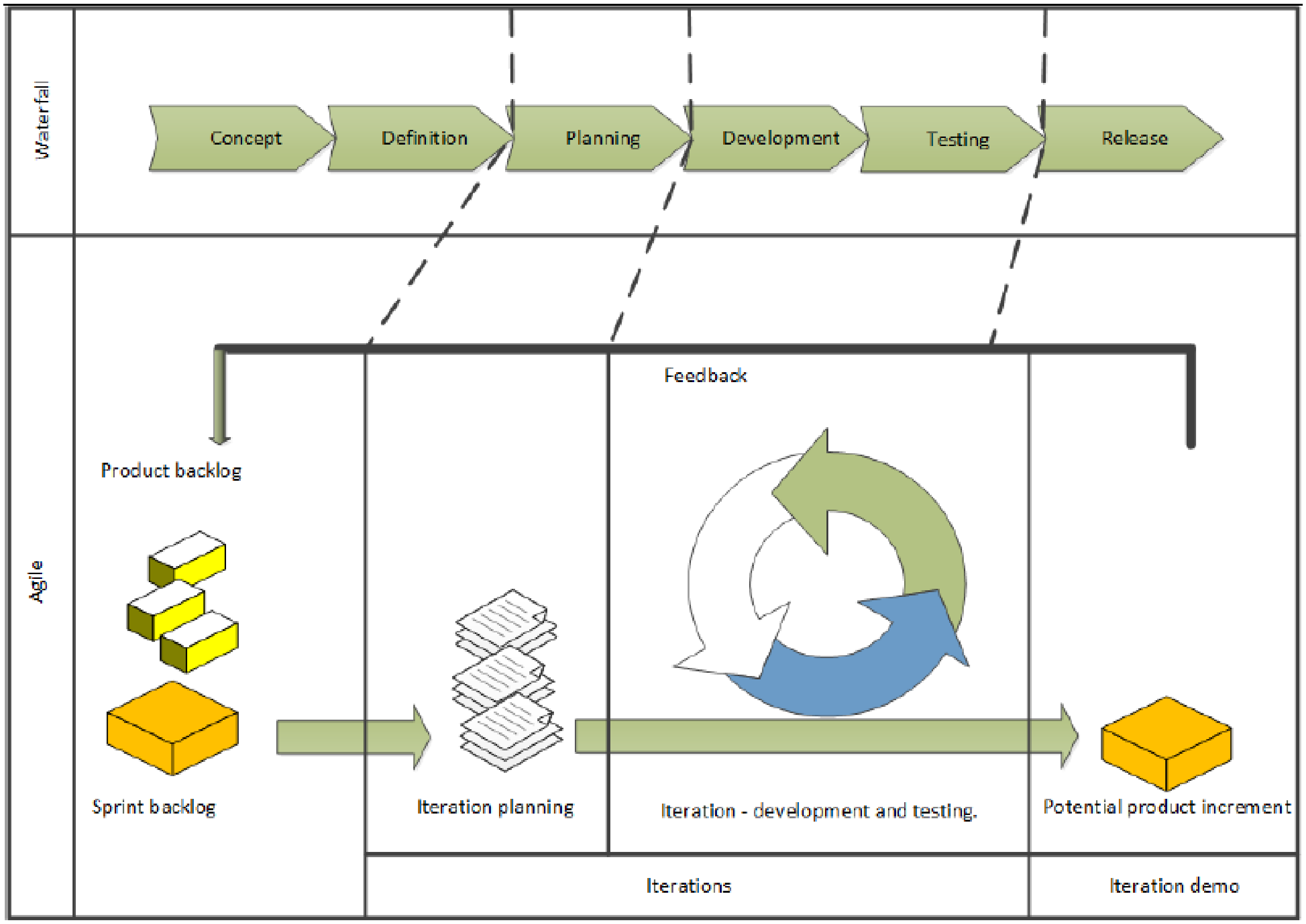

Fig. 6. Hybrid Development Model implemented at a Fortune 100 company (Mahadevan et al, 2015)

requirements and objectives, developers would not be able to develop/enhance/improve the system as users want[....]" "If the built software that would not help the business or process team to work, they would refuse working on the software. As the result, all the development time would be wasted."- Interviewee 1

"Very much, since the development team has to be clear with all the expectation of the final product." "....severity of issue, if the current needed features are crucial for the business and the changes does not require complete overhaul of the entire process then focusing on current needs should be [priority]."- Interviewee 2

"It is certainly important to nail down the scope the project early on but have some of the details open on requirements as long as possible. It is can be a bit of a balancing act between locking down requirements and being adaptive. [For instance,] Something like a workflow should be specified up front but the requirements in a notification email could wait until near the end of the process. Again flexibility is often tied to if something needs to be validated or not." - Interviewee 3

From the depicted statements, I could conclude that the software development team would need upfront product requirement and technical specification before they could start stepping into the project. Additionally, the 'Interviewee 1' stated that even when the requirement are loosely defined it helps them to "visualize end-users need". The requirements also provide them with a vision for the product (software) and how it might behave. Similar to the observations made in [2] [26] [23], upfront requirements can work as a blueprint to what product expectations are from the stakeholder. Meeting these "objectives" is considered as completing milestones; this sense of accomplishment positively influences the adoption of a development approach. The ability to add new elements as the project evolves is described as an advantage; it enables the team to re-prioritize their efforts to the urgent needs of the stakeholder. Thus, I infer that it is an advantage to have a defined conception phase (waterfall method) in the hybrid development method with an ability to modify development goals through collaborative discussions.

B. F2: Communication and Collaboration

Q2: How important is direct communication between team(s) members for defining requirement and development of the software? How do you share your work, ideas with other members?

“...each developer [is] coding on each module. However, we will discuss the process and technical development - high to low level - together.” “We will update each others progress frequently (almost daily) to ensure none of us are not stuck at certain area.” “We all agree with the solutions prior developer walks away to start coding his/her own module. The team will gain and share knowledge and understand the concept together..”- Interviewee 1

“[intra] team communication is important otherwise we might end up creating something which was not expected by the client.” “Yes, because one developer owns the process therefore he should.. [provide] clear picture of all the things he is working on”- Interviewee 2

“We do have a lot of knowledge transfer when it comes to these modules as the entire dev team is responsible for providing support after a module launches.” “I strongly believe in direct communication between teams. Simply having requirements on paper can often create unnecessary time wasted going back and forth. However, documented minutes of direct communication is necessary to ensure that people don’t have selective memories down the line.”- Interviewee 3

From the depicted statements, I could conclude that the interviewee perceive the communication in hybrid model to be of utmost importance. The interviewee’s mention the need for discussing the process and technical development since each member is working on their individual part. The interviewee’s did not mention the need to keep daily meetings as it might cause interruption in their day-to-day work. The need for communication via virtual or in-person meetings is left to need by basis and arrangements are to be made proximate to the meeting. Communication and Collaboration is very important throughout the project; when the developers bring together their individual piece of work to be stitched together, it helps them to understand each others code and debug the script if necessary. It further promotes sense of trust and mutual understanding between team members. However, additional efforts are required to coordinate and integrate the fine-grained development tasks done by individual developer and may lead to some additional integration complexity. The role of team leader is necessary to establish communication between the stakeholder and team, and providing the needed recourse to accomplish the task. Thus, I infer that regular meetings and team collaboration (Agile method) though are necessary, they should be organized according to the needs of either the developers, the manager (or team leader) or the stakeholder to take the project to desired end point.

C. F3: Individual responsibility and Motivation

Q3: As stated discussed previously each developer works on his/her part individually, does it encourage you to do better and raise responsibility?

“We will update each other progress frequently (almost daily) to ensure none of us are not stuck at certain area.” “[it] raises more selective responsibility as each developer is responsible for coding on separate module”- Interviewee 1

“[since] one developer owns the [code] and responsible for fixing if bugs arises in the future... sometimes it is difficult and needs more effort to help others and also share information on my work....” Interviewee 2

“Absolutely. I think it allows us accomplish things efficiently without having to think of all the potential inter connectivity that may or may not happen in the future. I think it helps breed some creative solutions to issues when they arise. We are still a pretty tight knit team who often reviews issues together but developers working independently still have the chance to find their own potential solutions before bringing them to the table.”- Interviewee 3

“Yes I believe it does. Additionally, enhancements are often picked up by any member based on availability so everyone needs to have a good working knowledge of the module.”- Interviewee 4

From the depicted statements I could conclude that interviewees did agree they are more mindful of their work since they need to help complete documentation as well as explain the code to others if required for debugging at the time of integration. The interviewee sometimes find it ‘difficult’ to continuously fulfill all their tasks and simultaneously support others. The higher self-responsibility required in hybrid approach “helps breed some creative solutions to issues when they arise.” However, in this approach the team “still [is] a pretty tight knit who often reviews issues together”. Thus, I infer that in hybrid method the individual requires more commitment to perform at desired level. Individual needs to collaborate with other team members so that their work is understood and to get feedback on their work. As stated by interviewee 3 and described in the [3] the development is lead by a manager or team leader who guides the teams, and there responsibility is to encourage the team and keep them motivated. Individual efforts (Agile) lead to cultivation of innovative ideas, thus, improving overall quality of work and commitment towards the desired goal.

D. F4: Documentation-Transferable knowledge and learnability

Q4: How would you describe the process of documentation in your organization and how important is it for organization? How does the documentation help a new employee?

“extremely critical for a new employee to learn and start working on a software project... developers would go through the training cycle of the company; even though the concept is similar across the industry. However, the process is unique for each company.”- Interviewee 1

“[Documentation] practice is obvious in our organization since we follow it strictly, spending a good amount of time doing [it after] all the individual stages.” “Since a lot of time is spent doing each activity I believe it should give enough time for the new employee to learn [through observation and from studying documentation.]”- Interviewee 2 “documented minutes of direct communication is necessary to ensure that people don’t have selective memories down the line.” “I think that the learnability of our development process is fairly easy. We have had several co-ops and members of different levels join our team and they have all been able to pick up on the process fairly easily [with practice and reading documented work]”- Interviewee 3

“...Besides it is more important for us in [medtech] company to keep detailed and complete documentation for each process..... We need to be prepared for external audits all [the] time because we are [serving] in critical market and documentation needs to be up to date.”- Interviewee 4

As depicted in the above statements, the interviewee stated that documenting the development process was of utmost importance in their organization for transferring knowledge within the organization and to support the new employee in the getting used to work style and learn about the development process. Since the organization operates in MedTech business, the System developer team is required to maintain up-to-date documentation and be prepared for an internal or external audit at all time. Systems Project Managers is responsible for documenting requirements and changes, project workflows, resources utilization, and expected system behavior. It is possible that the requirements change during the project life cycle, meanwhile, documentation is important to track changes in both requirements and functionality, and maintain a detailed audit trail. Thus I infer that documentation (Waterfall) is really crucial in hybrid development model implementation. The organization is able to efficiently transfer knowledge between teams, use it for helping new employees adopt to new process, and maintain an evidence for audit if needed by the organization.

E. F5: Cross-team roles

Q5: How do people work across different teams? Do you share resources? Who is responsible for handling cross team roles and resource sharing?

“Being flexible allows for some give and take negotiations and has generally been successful for us. I need to work with other managers to manage shared resources such as costs, technology investments. In specific scenario, other team does ask for supportive roles that are required if any employee is unavailable... developer is also asked for his ability to support along with their own work” “Cross team roles are not requested much, one time I can recall was when sale[****]ce administrator decided to leave the company then we had to support the development and maintenance of the platform for few weeks” “We supported sale[****]ce platform for almost a month when one of our colleagues decided to switch company....it was steep learning curve for a week, even after that we faced roadblocks in completing the request on time”- Interviewees

As depicted in the statements by the interviewees the cross team roles are necessary to communicate between teams and provide help the to cross team members if called for assistance. The Interviewee described cross team roles to be harder to understand and difficult to cope considering they still need to complete their own work. The managers and team leaders are responsible for cross team communication and collaboration. The interviewees stated that the cross team roles are requested only for short duration and these requests are very limited. Cross team roles are seen as an obstacle and tend to reduce team productivity. Thus I infer that cross team roles should not be promoted for use in hybrid development approach as it decreases team productivity and may lead to additional lead time due to steep learning curve.

F. F6: Flexibility and Project completion

Q6: How does the flexibility provided by the hybrid practices during the development impact productivity and time to deployment?

“We had created flexible schedule so that we can extend running the test script up to 16 hours a day instead of traditional schedule 9 to 5 that would be less than 8 hours running time.” “flexibility practices would not only lead to better productivity and better economic results, it also increases significantly to our customer satisfaction...having the flexibility [in] practice , we always can offer the shortest duration for each enhancement to our end users.” “Discussion will take place when there is any issue within 24 hours. This flexibility has a significant improvement for the team work environment within the organization.” “We see improvement in delivery time..... we deploy parts of the project 3-4 times a year and timeline is adjusted if needed......we begin with deadlines and dates for deployment to prod [production] environment but its little flexible.”- Interviewee 1

“Yes, I agree flexibility is good for delivery and to complete project....we tend not to change delivery time unless there were major changes in requirements from the user which requires a complete overhaul in system... sometimes it impacts the project completion ” “No, I have not seen a complete project failure in my experience here.....may be because we are not a software focused company...”- Interviewee 2

“Not always. Something that is late in the game for testing purposes can really throw validation efforts for a loop but we sometimes accept those headaches to ensure we can ensure internal customer satisfaction.”- Interviwee 3

“Yes. I feel that being able to be flexible very much fosters can do attitudes and happier team collaborations. Internal customer satisfaction at launch is what makes my day. Being flexible allows for some give and take negotiations and has generally been successful for us...I think it does lead to better productivity overall”- Interviwee 4

From the above depicted statements I can conclude that the flexibility (Agile) which hybrid development method bring to the project does have a positive impact on project completion and higher consumer satisfaction . The flexibility to make changes to requirements by stakeholders may add a little delay to final product release, but as stated by interviewees the project tends to get completed with low chances being completely scrapped. Since the interview was conducted in an Medtech SME with less focus on software, I am assuming that hybrid model does keep the chances of complete failure of the project low, however, considering the fact that a large organization would have multiple projects deployed across distributed location, the chances of failure might not be as minimal as the statements suggest, so it might require further work. Therfore, I infer that hybrid approach does improve project deliverability, meeting project deadlines, and keep failure of entire process low compared to waterfall method where entire project needs to be scrapped. The ability to be flexible and bring new enhancements “fosters can do attitude”, thus improving overall quality of work.

VIII. Conclusion and Future Work

Although hybrid agile-waterfall methods is widely used development approach in the industry, academic research has been lacking on the factors that determine their adoption and how these factors can affect the work efficiency in the organization. This paper does a detailed qualitative study on the advantages and disadvantages of both traditional waterfall and agile development methods, followed by characteristics that are generally visible in a hybrid development method. This is followed by a detail study of the hybrid approach which has been put in practice in various industries such as manufacturing, computer networks, software development, finance and banking etc. I carefully analyzed case-studies undertaken at multiple organizations from some of the sectors mentioned above and identified the factors and issues which impact the adoption of the hybrid development method. Using this information I identified combination of 6 factors that may impact adoption of hybrid development approach at an organization.

In my qualitative study I conducted interviews and survey questionnaires with the system developer team at a medtech company in Canada. Using the knowledge gained through the case-studies and qualitative research I deliberated on each factor that was identified in the case-studies. Factors F1, F2, F3, F4, F6 support the adoption of hybrid model, while, factor F5 has a negative impact. The current study has been limited in extent of interviews and survey questionnaire, because of time constraint and human resource.

In future I would like to conduct a combined qualitative and active research which covers a wide range of interviews and surveys to validate the implementation of hybrid development in broader area. The future work should explore how human interaction and social practices in an organization would impact development practices. The future study should explore inter-dependencies between the factors and should be able to provide a generalized view of the implementation of the hybrid approach for small-large organizations. With this paper, I hope I was able to explore and present points that would help promote more research in this field and provide understanding to readers for adoption of hybrid development approach.

Acknowledgment

I would like to thank Dr. Daniel M. Berry for his continuous support and feedback throughout the project. In addition, I would also like to thank the members of System Development team at a Canadian MedTech company who participated in the qualitative study for this project.

References

[1] Author Info Mendix May 15. Promoting agile in a waterfall culture, Jun 2019.

[2] Olayele Adelakun, Robert Garcia, Ted Tabaka, and Redar Ismail. Hybrid project management: Agile with discipline. 2017.

[3] Sanjiv Augustine, Bob Payne, Fred Sencindiver, and Susan Woodcock. Agile project management: Steering from the edges. *Commun. ACM*, 48:85–89, 01 2005.

[4] Daniel M Berry et al. From contract drafting to software specification: Linguistic sources of ambiguity-a handbook version 1.0. 2000.

[5] Jean Binder, LI Aillaud, and Lionel Schilli. The project management cocktail model: An approach for balancing agile and iso 21500. *Procedia-Social and Behavioral Sciences*, 119(119):182–191, 2014.

[6] Barry Boehm and Richard Turner. Management challenges to implementing agile processes in traditional development organizations. *IEEE software*, 22(5):30–39, 2005.

[7] Kathleen B Hass. The blending of traditional and agile project management. *PM world today*, 9(5):1–8, 2007.

[8] Hans Jonasson. *Determining project requirements*. Boca Raton : Auerbach Publications, 2008.

[9] Woi Hin Kee. Future implementation and integration of agile methods in software development and testing. In *2006 Innovations in Information Technology*, pages 1–5. IEEE, 2006.

[10] Rob J. Kusters, Youri van de Leur, Werner G. M. M. Rutten, and Jos J. M. Trienekens. When agile meets waterfall - investigating risks and problems on the interface between agile and traditional software development in a hybrid development organization. In *ICEIS*, 2017.

[11] Dean Leffingwell. *Scaling software agility: best practices for large enterprises*. Pearson Education, 2007.

[12] Lakshman Mahadevan, William J Kettinger, and Thomas O Meservy. Running on hybrid: Control changes when introducing an agile methodology in a traditional” waterfall” system development environment. *CAIS*, 36:5, 2015.

[13] Likoebe Maruping, Viswanath Venkatesh, and Ritu Agarwal. A control theory perspective on agile methodology use and changing user requirements. *Information Systems Research*, 20:377–399, 09 2009.

[14] Martin McHugh, Fergal McCaffery, and Valentine Casey. Barriers to adopting agile practices when developing medical device software. In *International Conference on Software Process Improvement and Capability Determination*, pages 141–147. Springer, 2012.

[15] M Sundararajan Murugaiyan and S Balaji. Succeeding with agile software development. In *IEEE-International Conference On Advances In Engineering, Science And Management (ICAESM-2012)*, pages 162–165. IEEE, 2012.

[16] Sridhar Nerur and VenuGopal Balijepally. Theoretical reflections on agile development methodologies. *Communications of the ACM*, 50(3):79–83, 2007.

[17] Sridhar Nerur, RadhaKanta Mahapatra, and George Mangalaraj. Challenges of migrating to agile methodologies. *Communications of the ACM*, 48(5):72–78, 2005.

[18] Maria Paasivaara, Benjamin Behm, Casper Lassenius, and Minna Hallikainen. Large-scale agile transformation at ericsson: a case study. *Empirical Software Engineering*, 23(5):2550–2596, 2018.

[19] Sachar Paulus, Nazila Gol Mohammadi, and Thorsten Weyer. Trustworthy software development. In *IFIP International Conference on Communications and Multimedia Security*, pages 233–247. Springer, 2013.

[20] Rupali Pravinkumar Pawar. A comparative study of agile software development methodology and traditional waterfall model. 2015.

[21] Candice Quist. Benefits of blending agile and waterfall project planning methodologies. 2015.

[22] Mojdeh Rahmanian. A comparative study on hybrid it project management. *International Journal of Computer and Information Technology*, 3(05):1096–1099, 2014.

[23] DeLine Researcher. Software development at microsoft observed: It’s about people … working together. 2005.

[24] Darrell K Rigby, Jeff Sutherland, and Hirotaka Takeuchi. Embracing agile. *Harvard Business Review*, 94(5):40–50, 2016.

[25] Sebastian Schlauderer, Sven Overhage, and Björn Fehrenbach. Widely used but also highly valued? acceptance factors and their perceptions in water-scrum-fall projects. 2015.

[26] Günther Schuh, Eric Rebentisch, Michael Riesener, Frederic Diels, Christian Dölle, and Steffen Eich. Agile-waterfall hybrid product development in the manufacturing industry—introducing guidelines for implementation of parallel use of the two models. In *2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pages 725–729. IEEE, 2017.

[27] A Sixsmith, G Mooney, and C Freeburn. Project management in practice: Views from the trenches. In *Proceedings of the 24th International Business Information Management Association Conference-Crafting Global Competitive Economies: 2020 Vision Strategic Planning and Smart Implementation*, 2014.

[28] Mario Špundak. Mixed agile/traditional project management methodology–reality or illusion? *Procedia-Social and Behavioral Sciences*, 119(1):939–948, 2014.

[29] Joseph S Valacich, Joey F George, and Joseph S Valacich. *Modern systems analysis and design*. Pearson Education Limited London, 2017.