International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 12 Issue: 2
Article Received: *25 December 2023* ***Revised:*** *12 January 2024* ***Accepted:*** *20 February 2024*

A Closer Look at Hybrid Agile Implementation in China

Li Yilin
***Siti Sarah Maidin**
Faculty of Data Science and Information Technology
INTI International University
Nilai, Malaysia
i22022709@student.newinti.edu.my
sitisarah.maidin@newinti.edu.my

***Norzariyah Yahya**
Normi Sham Awang Abu Bakar
Department of Computer Science,
Kulliyyah on Information Communication and Technology, International Islamic University Malaysia
Kuala Lumpur, Malaysia
norzariyah@iium.edu.my
nsham@iium.edu.my

Abstract—A software project must be developed based on specific software development models or frameworks such as the Waterfall model, agile methodologies or any suitable development approaches as needed by the software development team. The need to refer to a model or framework is to ensure that the project is systematically managed, the progress is monitored, and a project meets the requirements with a good quality project that is delivered to the end-users on time. Hybrid agile is a software development lifecycle model currently widely used in a software project. Waterfall and Scrum are two examples of a well-known hybrid agile model. Some software projects even use three combinations of a model that derives a hybrid agile model called Water-Scrum-Fall. Despite many successful stories on the hybrid agile implementation and a list of benefits of the project implementing hybrid agile being reported, it is unfair to make a generic assumption that hybrid agile is acceptable by the majority of software projects. This research explores another perspective of hybrid agile implementation in one East Asian country, China. This research adopted semi-structured interviews to explore the perspective of China's software engineering team on the hybrid agile implementation, and thematic analyses is used as the analysis approach. The finding shows that the software engineering team is inclined towards hybrid agile but acknowledges the advantages of hybrid agile.

Keywords-*Hybrid Agile; Perspective; China, Process Innovation, Industrial growth*

I. INTRODUCTION

Over the years, software development models and approaches have evolved tremendously. From the Waterfall model, introduced in 1970, to Agile in 2001, software development models have continued to grow to meet software project needs. Recently, the hybrid model has become one of the preferable models in a software project. Later, the DevOps movement bridged the gap between development and operations, emphasising continuous integration and deployment. As complexity increased, hybrid approaches that combined traditional and Agile practices gained interest in software projects. In today's landscape, adaptability, cooperation, and customer-centricity are championed in software development.

With single or hybrid models have been suggested, the implementation of development models exhibits variation among countries and teams. The efficacy of the development model employed directly influences the quality of the entire development project. Consequently, a significant body of research has investigated the prevailing software development patterns and methodologies [1][2][3]. Cusumano, MacCormack, Kemerer, & Crandall [1] conducted a preliminary descriptive study to examine the unique practices employed by software developers globally. The objective was to find the performance gaps associated with various development models utilised in software development across different countries. Furthermore, Kuhrmann et al. [2] examined various software development

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 12 Issue: 2
Article Received: *25 December 2023* ***Revised:*** *12 January 2024* ***Accepted:*** *20 February 2024*

methodologies and techniques, concluding that hybrid software development has emerged as a prevalent strategy in the field.

A single software development in a software project alone cannot accommodate the swift pace of software development. Hence, conducting a more comprehensive investigation into the hybrid Agile model, which integrates Agile methodologies with other development approaches to fulfil diverse project demands is necessary. Integrating many development models results from diverse methodologies' inherent benefits and drawbacks. This mixture of approaches is proposed as a feasible resolution, leveraging the strengths of one development model to mitigate the limitations of another approach. Hence, within the range of software development, there are model options, including the Waterfall, spiral, V-Model, Scrum, Lean, Kanban, and XP models. Additionally, the mixed agile model, which combines elements from various models, yields distinct outcomes that necessitate examination within specific contexts [4].

The software industry in China has experienced significant advancements in recent years, characterised by substantial increases in investment in software development and the ongoing formation of software firms. As a result, China has established itself as one of the leading software marketplaces globally [5]. Nevertheless, regarding technological advancements and the capacity for autonomous software development, China continues to trail behind the United States and other industrialised nations, as it remains in a development phase [6]. China's limited adoption of agile development approaches can be attributed to its collectivist values and adherence to conventional cultural norms [7]. According to Eriksson [8], China seeks to improve its software development capabilities.

Eriksson [8] made an insightful remark about agile practices in China. Many firms have adopted agile development methodologies to speed up innovation and adaptation. However, does China consider hybrid agile models suitable for the software engineering team? Is there any discrepancy that raises concerns of the software engineering team? Does a lack of understanding or a more profound cultural or managerial cause prevent Chinese enterprises from adopting hybrid practices? The proposed study seeks to answer these questions by investigating hybrid agile practices in Chinese software development businesses. This study seeks to understand China's software engineering team's perception of the hybrid agile adoption and company decisions towards hybrid agile.

II. Related Studies

The following topic deliberates the software development lifecycle (SDLC), then focuses on agile and specifically discusses Scrum, then the hybrid agile mode.

A. Software development life cycle

The Software Development Life Cycle (SDLC) offers a structured approach to developing and delivering software applications. The software industry uses the SDLC process to facilitate the design, development, and testing of software applications of superior quality. The Software Development Life Cycle (SDLC) is employed as a methodology to generate software of superior quality that aligns with client requirements, ensuring timely completion and adherence to estimated costs. The Software Development Life Cycle (SDLC) is a systematic approach that is adhered to by software organisations in order to manage and execute software projects.

The Software Development Lifecycle (SDLC) is a methodical and structured approach employed to strategise, construct, evaluate, and implement software applications. The significance of this resides in the assurance of timely delivery, adherence to budgetary constraints, and fulfilment of stakeholder expectations in software projects. By adhering to a well-defined lifecycle, software developers can proactively detect and resolve potential issues early, enhancing the product's overall quality and increasing productivity. The Software Development Life Cycle (SDLC) promotes effective stakeholder communication, enhances resource management, and guarantees ongoing improvement through iterative feedback. Adherence to the Software Development Life Cycle (SDLC) aids in the reduction of risks, the optimisation of costs, and the alignment of the final product with user requirements and corporate goals [9][10][11].

B. Agile development

Agile techniques distinguish themselves within software development by significantly emphasising a philosophy focused on collaboration and shared ideals. In contrast to conventional approaches, Agile methodology emphasises the cultivation of collaborative efforts and the expeditious attainment of client contentment. The process functions through rapid and succinct iterations, enabling stakeholders to assess developmental progress from the first phases. The approaches discussed in this context are rooted in the ideals outlined in the Agile Manifesto, with the primary objective being the efficient and prompt production of functional software [12][13]. The agile manifesto, which places significant emphasis on flexibility and interaction, has garnered substantial attention from developers, leading to the establishment of matching development communities and its widespread dissemination globally [14][15][16][17].

An Agile development model or framework refers to any development method that aligns with the principles outlined in the Agile Development Manifesto. At present, a selection of established agile frameworks includes the Extreme Programming (XP) model, Scrum model, Feature Driven Development (FFD) model, and Kanban model

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 12 Issue: 2
Article Received: *25 December 2023* ***Revised:*** *12 January 2024* ***Accepted:*** *20 February 2024*

[14][15][16][17]. In brief, all agile development models engage in iterative processes and prioritise prompt value delivery to the client. However, it is essential to note that each framework possesses its unique style and features within the Agile paradigm. Agile software development is an iterative process, wherein requirements and solutions are developed through collaborative efforts among self-organising cross-functional teams [10]. The agile methodology addresses some limitations of the waterfall paradigm to enhance the efficiency and effectiveness of the software development process. It does so by avoiding oversimplification and false expectations regarding requirements, hence promoting improved quality and speed. In contrast to the preceding process-oriented development, Agile development emphasises the interplay between developers and their interactions with customers, prioritising these relationships over contractual agreements. This principle is deeply ingrained in the fundamental values upheld by the development community [14]. The notion of flexible collaboration renders agile development a rapid iterative process characterised by intimate cooperation, adaptable adjustments, and immediate implementation.

1) Scrum

Scrum is an agile methodology utilised in software development, which significantly emphasises collaborative teamwork, individual accountability, and iterative advancement. Teams engage in time-boxed iterations known as fixed-length sprints, during which they strive to create increments that have the potential to be shipped. The Scrum framework fosters a culture of cooperation, adaptation, and continual feedback, ensuring the ongoing relevance and quality of the product throughout the development process. The Scrum development approach typically has a series of stages that commence with the Product backlog as shown in Figure 1. The iterative procedure involving numerous cycles is commonly called the Sprint cycle. The fundamental procedural stages of this approach will be primarily elucidated as follows [17]. There are three (3) fundamental roles in Scrum: Product Owner (PO), Scrum Master and Development Team. The Product Owner (PO) assumes the responsibility of representing stakeholders and overseeing the management of the Product Backlog. This entails ensuring that the team is focused on addressing the most critical issues.

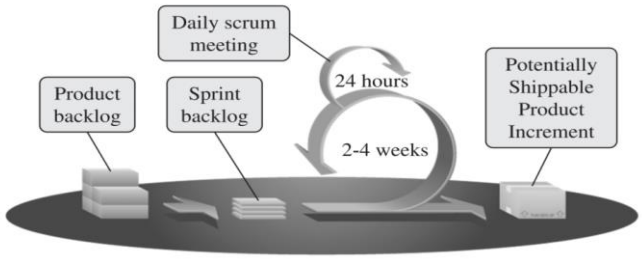

Figure 1 Scrum Process Dynamics [17].

The Sprint Backlog is a designated subset of the Product Backlog that is carefully chosen for the ongoing Sprint. It outlines the specific tasks that must be accomplished to achieve Sprint's overall objective. Increment refers to the deliverable item produced after a Sprint, intended to satisfy the team's predetermined criteria for completion. During the sprint planning phase, the team collectively determines the tasks and objectives that will be addressed and accomplished in the forthcoming Sprint. The Daily Scrum is a brief, 15-minute meeting held daily to facilitate coordination among development team members and provide a platform for addressing any obstacles or difficulties encountered during their work. The Sprint Review is a meeting after a Sprint, during which the team presents the completed work to the Product Owner and other relevant stakeholders. The Sprint Retrospective is a collaborative session in which the team reflects upon the previous Sprint and discusses potential areas of improvement for the subsequent Sprint. Implementing continuous feedback loops, such as Daily Scrum, Sprint Review, and Retrospective, facilitates the process of inspection and adaptation. These feedback mechanisms allow the team to assess their progress and make necessary adjustments in response regularly. Collaboration is a fundamental aspect of project management that places significant emphasis on fostering open communication channels, engaging in collective decision-making processes, and actively involving stakeholders at various stages of the project. Therefore, we can summarise that the iterative approach employed by Scrum facilitates the consistent provision of feedback, augmenting the ability to forecast project outcomes. The implementation of short sprints in project management aids in the effective management of risks by rapidly identifying and resolving difficulties as they arise. Through the strategic prioritisation of tasks of high value and the active involvement of stakeholders, the Scrum framework effectively facilitates the development and efficient delivery of the most valuable features of a software product over its entire lifecycle. The Scrum framework facilitates software development by employing a process of iterative and incremental development. While the concept does not inherently create loopholes, it is susceptible to certain flaws through implementation. Table 1 summarises a concise overview of common problems in scrum development.

TABLE 1 COMMON PROBLEMS IN SCRUM DEVELOPMENT

No	Problem	Reason	Reference
1	Lack of time to resolve all issues	Short release cycles	[18]
2	Lack of documentation for the planning phase	Scrum team knowledge monopoly	[19]

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 12 Issue: 2
Article Received: *25 December 2023* ***Revised:*** *12 January 2024* ***Accepted:*** *20 February 2024*

3	Lack of guidelines for requirements	There is no process for freezing requirements	[20]
4	Developers tend to lack security awareness	Too much pressure on speed	[18]
5	Vulnerabilities cannot be fixed in time	The security assessment in Sprint is not perfect	[18]

C. Non-Agile

The term non-agile is commonly used to denote the conventional software development processes, which are opposed to Agile methodologies such as Scrum. Non-agile methodologies, sometimes called plan-driven approaches, entail sequential stages: requirements gathering, design, implementation, testing, and maintenance. Each phase must be concluded before initiating the subsequent phase. Integrating changes becomes challenging once the project is in progress, rendering these approaches less flexible in accommodating evolving demands. The primary focus is placed on comprehensive documentation and rigorous adherence to the original design. Non-agile approaches demonstrate efficacy in projects characterised by well-delineated needs and limited alterations. However, they exhibit a deficiency in the fundamental attributes of flexibility and adaptability intrinsic to Agile methodology. Compared to agile development, non-agile development methods typically exhibit less lightweight and flexibility. On the contrary, they are better suited for projects of significant scale or those with well-defined requirements [21][22][23]. Waterfall project management, often known as the Waterfall development model, is a conventional approach to software development characterised by a linear sequential design process. It is considered a prominent exemplar of non-agile development methodologies. The foundational framework of the waterfall model was initially introduced in 1970, with the specific terminology "waterfall" being coined in 1976. Since its inception, the model has been subject to extensive scholarly examination, thus establishing its status as a prominent and enduring framework within software development life cycle (SDLC) methodologies [24].

The Waterfall model is often regarded as the most traditional and oldest Software Development Life Cycle (SDLC) model, representing the fundamental concept of SDLC [25]. The waterfall model, being a linear sequence model, entails the division of the software development process into a number of sequential stages. It mandates the completion of all tasks from the preceding phase before progressing to the subsequent phase [21]. While the waterfall model comprising six distinct components generally denoted as requirements acquisition, design, development, test, deployment, and maintenance. The waterfall model underscores the notion that software development is a methodical and rigorous undertaking, characterised by a sequential progression from high-level design to meticulous parameter planning. It necessitates thorough and standardised documentation, followed by a step-by-step execution of the planned activities.

Hence, although acknowledged for its numerous drawbacks, such as being unwieldy and inefficient, comprehending the incorporation of risk management and security in the development process of this model remains beneficial for exploring other interconnected SDLC models. Applying the risk management process at every stage of the Software Development Life Cycle (SDLC) can effectively guarantee the comprehensive security of the system and strike a balance between the organisation's asset requirements and security controls [9]. The literature review in Table 3 provides a comprehensive synthesis of studies exploring the integration of risk management in waterfall development. It summarises the safety measures and objectives associated with the significant stages of this development approach [9][25][26][27]. The risk management of the Software Development Life Cycle (SDLC) offers a systematic approach to security development within organisations. It encompasses a mindset that identifies, mitigates, and visualises risks [27][42].

D. Hybrid Agile

The early concept of hybrid agile development was introduced by [28] in his book "Adaptive Software Development." Highsmith [28] argued that Agile development must continuously adapt to technological advancements and proposed a hybrid development approach to address the increasing complexity of development requirements. The Hybrid Agile development technique encompasses the integration of various agile development methodologies or integrating Agile development with other development methodologies. This integration aims to enhance the overall development outcomes or cater to a broader spectrum of development requirements. The hybrid agile model is a synthesis of conventional approaches, typically the Waterfall approach, with Agile methodologies. The objective is to capitalise on the advantages of both approaches, accommodating the distinct needs of different projects. The hybrid agile approach effectively integrates the components of Waterfall and agile techniques. The waterfall methodology prioritises thorough documentation, careful initial planning, and a linear sequence of stages. In contrast, the Agile methodology encompasses critical aspects such as adaptability, iterative development, and a strong emphasis on regular feedback from stakeholders, which collectively contribute to the incremental delivery of products. In practical scenarios, it is common for teams to initiate their projects using the structured Waterfall

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 12 Issue: 2
Article Received: *25 December 2023* ***Revised:*** *12 January 2024* ***Accepted:*** *20 February 2024*

approach, particularly during the initial stages of requirements gathering and system design. This choice mainly influences the need for comprehensive documentation and adherence to strict regulatory standards. Nevertheless, after establishing this fundamental basis, there is a shift towards the Agile paradigm, namely in the development and testing stages. The rationale behind this transition is driven by Agile methodologies' inherent adaptability and ability to effectively respond to changing project factors. The advantages of employing this hybrid methodology are apparent in its adaptability, as it strikes a balanced compromise between the structured, sequential dependencies of the Waterfall approach and the iterative and flexible nature of Agile methodologies [28]. Additionally, incorporating the plan-driven development technique inside this model contributes to effective risk management. The utilisation of Agile methodologies facilitates proactive risk management, as it allows for the identification of potential hazards within the earliest planning phases. Agile's iterative and cyclic nature further enables the prompt resolution of these identified risks. Additionally, this model demonstrates proficiency in effectively handling the organised aspects of projects while being receptive to unforeseen demands or unanticipated obstacles. The selection of a hybrid agile model occurs when organisations desire the model provided by conventional approaches while still acknowledging the benefits of agility, particularly when project modifications are expected during its duration. The objective is to achieve optimal outcomes by combining the advantages of two distinct approaches but necessitating proficient administration for successful implementation. Furthermore, incorporating a combination of agile methodologies into the development process has the potential to enhance the attributes of rapid iteration and continuous integration, thereby ensuring a high level of quality while simultaneously boosting the efficiency of the development process [29]. Following prior discussions, hybrid agile development strategies frequently emerge as a fusion of diverse development approaches. Following this viewpoint, several researchers have proposed or evaluated several hybrid models [28][29][31]. To mitigate the risk of quality degradation in SCRUM's scheduling, [29] proposed an innovative hybrid methodology that combines SCRUM with Feature Driven Development (FFD), thus successfully reconciling the challenges about quality and scheduling in the Software Development Life Cycle (SDLC). This analysis specifically focused on the implications for transition costs and benefits. Gemino, Horner Reich & Serrador [31] conducted a recent empirical investigation wherein they surveyed 296 industry professionals, offering valuable insights. The researchers' findings provided a clearer understanding of the comparative consequences of traditional, agile, and hybrid agile methodologies on the success of project management, which finally shed light on the hybrid agile explorations. Various approaches to integrating development methodologies exist. However, empirical evidence from case studies indicates that agile development methods exhibit a commendable success rate of 71.5% [32]. Consequently, many organisations are embracing and implementing agile development practices. Consequently, integrating agile development with other development methodologies has reached maturity and prevalence. The term "hybrid agile development method" refers to a novel approach in software development that integrates agile development with either additional or non-agile development methodologies [28][29]. By considering the Scrum and waterfall models as illustrative examples, it is possible to integrate both to generate a novel hybrid model. The model referred to as the Water-scrum-fall (WSF) or scrumfall model has been documented by [33][34][35]. West, Gilpin, Grant, & Anderson [33] demonstrate the practicality and viability of the WSF model and argue that when the Scrum model, which has gained widespread popularity in agile development, encounters resistance from traditional development methods, the WSF model emerges as a proposed solution to this conflict and evolves. Figure 2 illustrates the concept of Water-Scrum-Fall development [33].

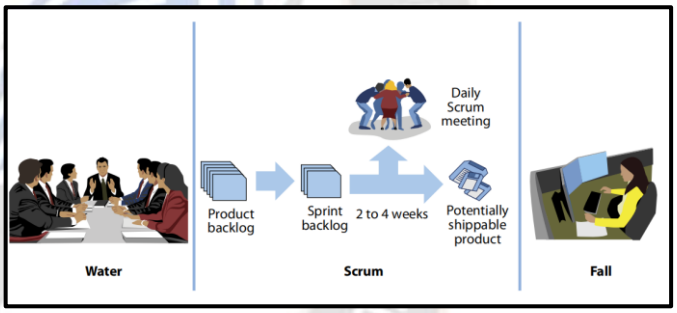

Figure 2 The concept of Water-Scrum-Fall development [33].

Numerous research endeavours have been conducted globally to investigate the viability and extent of hybrid agile models, resulting in the emergence of a substantial body of novel hybrid agile models. The viability and refinement of the WSF model are now being tested. Rahim, Chowdury, Nandi, Rahman, & Hakim [34] introduced a practice model known as "Scrumfall" that integrates the latter portion of the Waterfall model with the original Scrum model. This model demonstrated significant efficacy in managing large-scale projects and was subsequently implemented successfully within a company located in Bangladesh. In their study, Theocharis, Kuhrmann, Münch, & Diebold [35] conducted a comprehensive analysis of 22 pertinent literature sources. Their findings confirm that the Scrum model has emerged as the predominant development paradigm, while the WSF has successfully materialised due to the influence of several contributing variables. In conclusion, a collection of hybrid agile development models, such as the WSF model, have been actualised and have demonstrated noteworthy

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 12 Issue: 2
Article Received: *25 December 2023* ***Revised:*** *12 January 2024* ***Accepted:*** *20 February 2024*

value and promise. Hybrid agile models are projected to increasingly dominate the trajectory of software development in the future, assuming a heightened level of significance.

E. China Software Development Practices

The software development industry in China has exhibited consistent growth over an extended period and is now assuming a significant role in the nation's economic growth plan. China continues to exhibit a relative disparity in comparison to more advanced nations. However, it is making significant strides in narrowing this gap rapidly. Based on the findings presented in the China Software Industry Statistical Yearbook 2021, the revenue generated by China's software and information technology service industry in 2020 amounted to 8.16 trillion yuan, exhibiting a year-on-year growth rate of 13.3% [36]. The revenue generated by the software sector amounted to 7.15 trillion yuan, reflecting a 15.6% increase compared to the previous year. This figure accounted for 87.7% of the total revenue generated by the industry. In 2019, the total number of individuals employed in the Information Technology (IT) field amounted to 26 million, representing around 2.9 % of the overall workforce within the nation. According to data provided by the China Internet Network Information Centre in 2021, the Internet industry in China has had significant growth, with a total of 930 million Internet users, resulting in an internet penetration rate of 66.6%. Out of the whole population, mobile Internet users comprise 98.6%. Despite the aforementioned noteworthy figures, China must persist in fostering research and development and augmenting innovation to maintain a competitive edge. Additionally, China must be able to swiftly adjust to the evolving dynamics of the contemporary IT and software sectors [8].

The software development sector in China has consistently been a subject of investigation within the IT area, with numerous studies aiming to identify issues and generate novel insights from Chinese software development [5][8][37][38]. In their study, Wang and Li [37] conducted a thorough investigation of the prevalent quality issues in software development within China's software sector. As a result, they proposed a comprehensive solution and toolkit for software process management, aiming to address these challenges effectively. In a fundamental work, Eriksson [8] extensively examined the relationship between agile development and Chinese culture. The study revealed that agile development not only overcomes various challenges in the Chinese context but also aligns effectively with China's unique national conditions. Zhou, Khan, Liang, & Badshah [5] conducted a questionnaire survey on 34 developers across various regions in China. Their study effectively examines the prevailing practices within China's software industry during production and presents an initial summary of the country's software development and deployment procedures. The empirical investigation conducted on China's software sector yields practical remedies for China's specific context and offers valuable insights for other nations globally, particularly those in the developing category.

III. RESEARCH METHODOLOGY

This study employs a qualitative research approach and utilises semi-structured interviews as the primary data collection method. The participants in this study are software developers from China who have experience in software development, with a minimum of five years of professional experience. The utilisation of semi-structured interviews in this study is attributed to their inherent freedom in navigating the interview questions, hence providing many opportunities to delve further into the responses. The present study employed synchronous real-time communication through video conferencing, utilising a multichannel approach [39] The present study has chosen to utilise computer-mediated communication as a means of convenience in engaging with geographically dispersed participants [39]. Tencent has been chosen as the instrument for conducting the semi-structured interview for numerous reasons, including its user-friendly design, intuitive navigation, and its widespread adoption as the default programme for communication in China. Figure 3 illustrates the interview protocol used in this research.

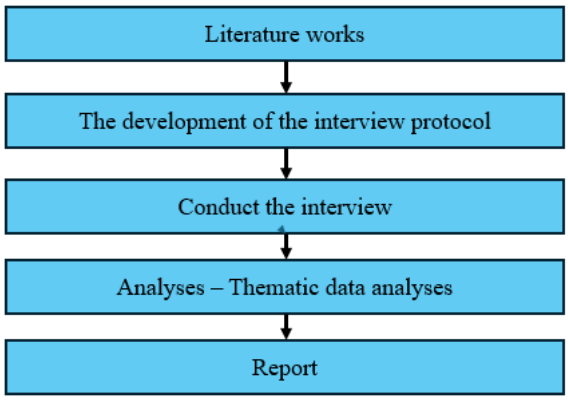

Figure 3 Research Design

The interviews aim to understand the execution of the Hybrid Agile development lifecycle by the software engineering process team, drawing on their expertise in Hybrid Agile software projects. The survey questions aim to examine the existing development approaches, techniques, and models employed by software engineers in their software development projects that result in the implementation of Hybrid Agile methodologies. Every interview will be documented by recording and transcribing the spoken content into written language. The examination of the interviews will rely on qualitative data analysis methodologies proposed by [40]. The material that has been transcribed will be systematically arranged according to the questions posed and then grouped into distinct themes. These

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 12 Issue: 2
Article Received: *25 December 2023* ***Revised:*** *12 January 2024* ***Accepted:*** *20 February 2024*

themes will serve as descriptive categories that capture the many components detected within the text [41]. Emergent categories, referred to as themes, concepts, or categories, will be identified from the transcribed text [40]. Identifying and interpreting patterns and linkages within the categories will ultimately be undertaken. The interview consists of three primary sections, each containing several sub-questions. Each section examines aspects that are relevant to the current development landscape, gradually exploring from a comprehensive overview to an in-depth analysis of developers' utilisation of development models. The initial portion assesses the predominant inclinations and viewpoints of developers in the process of model selection. The subsequent part explores the intricate approaches and ideas utilised by developers. The third portion employs a less prescriptive approach, seeking to ascertain the extent to which developers are now involved in or inclined towards the hybrid agile development methodology.

IV. Finding and Analysis

Thematic analysis is a qualitative research method that involves identifying, analysing, and reporting patterns, known as themes, within the data. This approach comprehensively explains participants' experiences and perspectives, generating valuable insights. ATLAS.ti facilitates the process by aiding in the organisation of data, supporting the production of codes, visualising linkages within the data, and managing the iterative process of refining and linking codes to emergent themes. Three (3) themes were identified from the interview. The themes are the selection of the software development model, current practices of development phases and perception of the Hybrid Agile model.

A. Theme 1: Selection of Software Development Model

According to respondent 1, the predominant development models utilised by their team primarily consist of prototype models. This preference stems from the prevalence of software projects focused on creating Manufacturing Operation Management (MOM) systems. Since each MOM vendor employs distinct production and management models and considerable uncertainty surrounding customer requirements, prototype models are deemed most suitable. Hence, in software design, it is imperative to prioritise the genuine requirements of customers as the initial step, elucidate the scope of these requirements, and develop prototypes that cater to the many business situations of customers. Nevertheless, respondent 1 expressed their intention to employ a prototype development model for uncomplicated and modest projects while opting for a waterfall or spiral development model for extensive projects to handle intricate modifications. For respondent 2, the waterfall model is employed in cases where a project undergoes extensive investigation over an extended duration, involving the presence of a professional requirements individual and the development of a comprehensive requirements document. In contrast, agile approaches are well-suited for projects characterised by dynamic and evolving requirements that span a significant duration. Various budgets and types of software projects employ distinct methodologies during the implementation process, making it challenging to fully adhere to the software engineering process or specifications in real-world projects. Hence, it is imperative to align the chosen development model with the nature of the project rather than its scale. Respondent 3 has implemented the Agile Scrum model because their team prefers a compact project structure. The Scrum model is particularly advantageous as it rapidly delivers an initial version of the product's infrastructure to end-users. As per the assertion made by Respondent 3, in the case of a significant or moderate-scale project, it may be imperative to conceptualise and assess the project's comprehensive framework during its first stages. Subsequently, an appropriate model should be selected per the project's specific requirements and characteristics. Prototyping is a feasible approach; nonetheless, adhering to the overarching architectural framework during the prototyping process is imperative.

Theme 2: Practices of development phases

Respondent 1 asserted that their team diligently adhered to the prototype development approach, iteratively refining and enhancing the development process based on the findings from prior demand surveys. This iterative process continued until the prototype's functionality aligned with the specific requirements of the customer's present real-world business environment. As per the assertion made by the first respondent, the process of requirements collecting has significant importance in executing development activities throughout a project's life cycle. This is due mainly to its profound influence on all future developmental procedures. Regarding the implementation of development, Respondent 1 asserts that design holds greater significance as it directly influences the programme's overall quality. This prioritisation remains consistent even when faced with impending deadlines.

Respondent 2 elucidates that his team intends to employ the waterfall development approach to implement their agricultural machinery knowledge base. It is acknowledged that the ultimate impact of this development may vary among individuals. According to respondent 2, project planning is seen as the pivotal stage in the actual execution of development since any misstep in this phase can lead to a misguided trajectory for the entire project. Respondent 2 asserts that the crucial stages of growth do not possess a universally applicable solution but rather are contingent upon the client's unique demands. Respondent 2 adheres to a predetermined time allocation for each phase of the development process and maintains consistent prioritisation and

time allocation for each phase, even when faced with approaching deadlines.

Respondent 3 expresses a high level of assurance over his team's comprehensive implementation of the Scrum model. Furthermore, he emphasises the significance of the product manager's involvement in requirements collecting and analysis as a pivotal aspect of this process. During the implementation phase of development, the design aspect holds utmost significance for Developer 3's team. This is due to its direct influence on the quality of the final product. Consequently, the priority of design remains unchanged, even in the face of approaching deadlines.

Theme 3: Knowledge of the Hybrid Agile Model

Respondent 1 has no Agile development experience but has some observational evidence about hybrid Agile development approaches. As emphasised by Respondent 1, hybrid agile development commonly employs an "iterative development" framework, wherein the software project requirements are divided into many iterations. Each iteration undergoes development, testing, feedback, and subsequent delivery. Respondent 1 posits that this model can compensate for the deficiencies inherent in a singular development model, such as the waterfall development approach. Waterfall development, in its essence, emphasises a singular role. However, when employed in conjunction with agile development, team members can foster trust, provide assistance, and collaborate harmoniously, optimising the utilisation of resources in an effective manner. Respondent 1 posited the possibility of a dearth of development documentation and a potential decrease in collaboration within the operational team.

Respondent 2 expressed their willingness to experiment with this approach, contingent upon the availability of a comprehensive and well-structured instructional guide. In the context of projects necessitating prolonged maintenance and upgrades, the preference leans towards adopting the waterfall methodology over the Agile style. The waterfall model can efficiently facilitate the development of projects that can be readily deployed in a production setting, thereby effectively addressing the immediate requirements of the user. As the system is utilised, users can consistently modify business processes or business models, enabling ongoing enhancements to the product through the implementation of agile methodologies.

Respondent 3 has not engaged in implementing the hybrid Agile model; nevertheless, they believe that this approach remains focused on meeting requirements and can manage projects that encompass agile and traditional methodologies effectively. Accordingly, the assertion made by Respondent 3 posits that the use of a hybrid agile development paradigm has the potential to enhance the efficiency of software delivery processes and mitigate project expenses. In the context of intricate and extensive software systems, respondent 3 contends that the hybrid agile development model poses challenges in accurately determining all the necessary tasks at the outset of the software development process. Consequently, if the customer lacks a comprehensive understanding of their desired end result, the quality of the project may be compromised upon completion.

V. Discussion and Conclusion

This research aims to examine the implementation of software development models inside Chinese software development firms, with a specific focus on the comprehension and utilisation of Hybrid Agile by their developers. This study critically examines the enormous body of literature on diverse software development methods, such as Scrum and Waterfall approaches, with the overarching objective of identifying and comprehending the key elements of Hybrid Agile. The hybrid Agile approach integrates elements from both the Scrum and waterfall models. It has undergone evaluation and has demonstrated its ability to effectively merge the benefits of precise requirement gathering and prompt customer feedback. This study employed interview techniques and conducted semi-structured interviews with a sample of three experienced software developers from China. This study employs thematic analysis to identify and analyse key findings derived from the perspectives of Chinese participants regarding hybrid agile practices. Chinese participants exhibit a greater propensity for utilising non-agile methods, with a majority believing that the selection of a development model is not contingent upon the scale of development. Participants in China exhibit a tendency to comprehensively execute a singular model throughout each stage of the project. The present study revealed that a lack of appropriate development experience for hybrid agile models was observed across all participants in China. Chinese participants acknowledged the benefits associated with the Hybrid Agile methodology; nevertheless, their inclination towards adopting this approach remained unbiased. This study has answered the first research question, demonstrating that non-agile models (such as Waterfall) and agile models (such as Scrum) are employed in China, but Chinese developers prefer the non-hybrid agile model. The study also sheds light on the second research question inquiry regarding the viewpoint of Chinese developers on hybrid Agile development. Despite recognising its advantages, there is a discernible reluctance to embrace it.

References

[1] M. Cusumano, A. MacCormack, C.F. Kemerer, and B. Crandall, "Software development worldwide: The state of the practice," IEEE software, 20(6), pp. 28-34, 2003.

[2] M. Kuhrmann, P. Diebold, J. Münch, P. Tell, V. Garousi, M. Felderer, K. Trektere, F. McCaffery, O. Linssen, E.

Hanser, and C.R. Prause, "Hybrid software and system development in practice: Waterfall, Scrum, and beyond," Proceedings of the 2017 International Conference on Software and System Process, 2017, pp. 30-39.

[3] N. Kurapati, V.S.C. Manyam, and K. Petersen, "Agile software development practice adoption survey," in Agile Processes in Software Engineering and Extreme Programming: 13th International Conference, XP 2012, Malmö, Sweden, Springer Berlin Heidelberg, 2012, pp. 16-30.

[4] N. Yahya, and S.S. Maidin, "The Waterfall Model with Agile Scrum as the Hybrid Agile Model for the Software Engineering Team," In 2022 10th International Conference on Cyber and IT Service Management (CITSM), IEEE, 2022, pp. 1-5.

[5] P. Zhou, A.A, Ali Khan, P. Liang, and S. Badshah, "System and software processes in practice: Insights from chinese industry," In Evaluation and Assessment in Software Engineering, 2021, pp. 394-401.

[6] S. Hermes, E. Clemons, M. Schreieck, S. Pfab, M. Mitre, M. Böhm, M. Wiesche, and H.S. Krcmar, "Breeding Grounds of Digital Platforms: Exploring the Sources of American Platform Domination, China's Platform Self-Sufficiency, and Europe's Platform Gap," 2020.

[7] X. Qiao, "Analysing the impact of Chinese cultural factors on agile software development," 2018. (Master's thesis).

[8] J. Eriksson, "Analysing potential barriers of Agile adoption in Chinese software development organisations," 2015.

[9] S. Radack, "The system development life cycle (sdlc)," (No. ITL Bulletin April 2009 (Withdrawn)), National Institute of Standards and Technology, 2009.

[10] J. Salokhidinov, "A comparison between agile (incremental), non agile (predictive) and DevOps methods in global software development, 2022. (Doctoral dissertation, Politecnico di Torino).

[11] F. Ahmed, L.F. Capretz, and P. Campbell, "Evaluating the demand for soft skills in software development," IT Professional, 14(1), 2012, pp. 44-49.

[12] R. Martin, Agile Software Development, Principles, Patterns, and Practices, Prentice Hall: Upper Saddle River, NJ, USA, 2002.

[13] R. Martin, Clean Code: A Handbook of Agile Software Craftsmanship, Pearson Education: London, UK, 2008.

[14] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta, "Agile software development methods: Review and analysis," arXiv preprint arXiv:1709.08439, 2017.

[15]K. Beck, "Embracing change with extreme programming," Computer, 32(10), 1999, pp.70-77.

[16] A.F. Chowdhury, and M.N. Huda, "Comparison between adaptive software development and feature driven development," in Proceedings of 2011 International Conference on Computer Science and Network Technology, IEEE, Vol. 1, 2011, pp. 363-367.

[17] B.V. De Carvalho, and C.H.P Mello, "Scrum agile product development method literature review, analysis and classification," Product: Management and Development, 9(1), pp. 39-49.

[18] I. Ghani, Z. Azham, and S.R. Jeong, "Integrating software security into agile-Scrum method," KSII Transactions on Internet and Information Systems (TIIS), 8(2), 2014, pp. 646-663.

[19] K. Brady, "AGILE/SCRUM Fails to get to grips with Human Psychology," 2006.

[20] X. Ge, R.F. Paige, F. Polack, and P. Brooke, "Extreme programming security practices," in Agile Processes in Software Engineering and Extreme Programming: 8th International Conference, XP 2007, Springer Berlin Heidelberg, 2007, pp. 226-230.

[21] M. McCormick, "Waterfall vs Agile methodology," MPCS, N/A, 3, 2012.

[22] M. Kramer, "Best practices in systems development lifecycle: An analysis based on the waterfall model," Review of Business & Finance Studies, 9(1), 2018, pp. 77-84.

[23] B. Boehm, and W.J. Hansen, Spiral development: Experience, principles, and refinements, Carniegie-Mellon University Pittburgh PA Software Engineering Institue, 2000.

[24] T.E. Bell, and T.A. Thayer, "Software requirements: Are they really a problem?," in Proceedings of the 2nd International Conference on Software Engineering. 1976, pp. 61-68.

[25] S. Balaji, and M.S. Murugaiyan, "Waterfall vs. V-Model vs. Agile: A comparative study on SDLC," International Journal of Information Technology and Business Management, 2(1), 2012, pp. 26-30.

[26] C. Banerjee, and S.K. Pandey, "Software security rules, SDLC perspective," arXiv preprint arXiv:0911.0494, 2009.

[27] K. Sahu, R. Shree, and R. Kumar, "Risk management perspective in SDLC," International Journal of Advanced Research in Computer Science and Software Engineering, 4(3), 2014.

[28] J. Highsmith, Adaptive software development: a collaborative approach to managing complex systems, Addison-Wesley, 2013.

[29] S.S. Tirumala, S. Ali, and A. Babu, "A hybrid agile model using SCRUM and feature-driven development," International Journal of Computer Applications, 156(5), 2016, pp.1-5.

[30] R. Wankhede, "Hybrid Agile Approach: Efficiently Blending Traditional and Agile Methodologies," 2016.

[31] A. Gemino, B. Horner Reich, and P.M. Serrador, "Agile, traditional, and hybrid approaches to project success: is hybrid a poor second choice?," Project Management Journal, 52(2), 2021, pp. 161-175.

[32] L. Neelu, and D. Kavitha, "Estimation of software quality parameters for hybrid agile process model," SN Applied Sciences, 3, 2021, pp. 1-11.

[33] D. West, M. Gilpin, T. Grant, and A. Anderson, "Water-scrum-fall is the reality of agile for most organisations today," Forrester Research, 26, 2011, pp. 1-17.

[34] S. Rahim, A.E. Chowdhury, D. Nandi, M. Rahman, and S. Hakim, "ScrumFall: a hybrid software process model," International Journal of Information Technology and Computer Science, 10(12), 2018, pp. 41-48.

[35] G. Theocharis, M. Kuhrmann, J. Münch, and P. Diebold, "Is water-scrum-fall reality? on the use of agile and traditional development practices," in Product-Focused Software Process Improvement: 16th International Conference, PROFES, Bolzano, Italy, Springer International Publishing, 2015, pp. 149-166.

[36] Government of the People's Republic of China, 2021 Statistical Communique on the Software and Information Technology Services Industry [Press release], 2022, Retrieved from http://www.gov.cn/xinwen/2022-01/28/content_5670905.htm (L. Meng, Ed.)

[37] Q. Wang, and M. Li, "Software process management: Practices in China," in Unifying the Software Process Spectrum: International Software Process Workshop, SPW 2005, Beijing, China, Springer Berlin Heidelberg, 2006, pp. 317-331.

[38] A.A. Khan, M.Nadeem, and M.A. Akbar, "Agile trends in Chinese global software development industry: Fuzzy AHP based conceptual mapping," Applied Soft Computing, 102, 107090, 2021.

[39] J. Salmons, "Designing and Conducting. Cases in online interview research," 2011.

[40] M. Renner, and E. Taylor-Powell, "Analysing qualitative data. Programme Development & Evaluation," University of Wisconsin-Extension Cooperative Extension, pp. 1-10.

[41] P. Bazeley, "Analysing qualitative data: More than 'identifying themes'," Malaysian Journal of Qualitative Research, 2(2), 2009, pp. 6-22.

[42] W. Y. Wong, T. H. Sam, C. W Too. and A. A. Khin, "Critical Success Factors of Operational Excellence in Software Quality Assurance: Best Practices for Integrated Change Control Management," 2023 19th IEEE International Colloquium on Signal Processing & Its Applications (CSPA), Kedah, Malaysia, 2023, pp. 287-291, doi: 10.1109/CSPA57446.2023.10087552.