Universiteit Leiden

ICT in Business and the Public Sector

Developing a Customised Agile Methodology for AI and Machine Learning Projects

Nipuni Thisarangi Dissanayake
Student No: s3223744

1st supervisor: Dr. Marc Hilbert
2nd supervisor: Dr. C.J. Stettina

Date: 21/08/2023

MASTER'S THESIS

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

Introduction: The research project aims to develop a customised Agile methodology for Artificial Intelligence (AI), Machine Learning (ML) and Data Science (DS) projects. Although Agile approaches are increasingly popular in software development, managing and executing AI/ML and DS projects can be challenging due to their unique characteristics, such as complexity, experimentation, and rapid iteration. Current Agile methods still need to accommodate the needs of AI/ML and DS projects. Even with alternative frameworks available, seamlessly combining multiple software development methodologies can be difficult for organisations. To address these challenges, creating a customised Agile methodology specifically for AI/ML and DS projects can help manage the challenges typically associated with AI/ML projects.

Methodology: To develop a customised Agile methodology for AI/ML and DS projects, the Design Science Research (DSR) approach was utilised. Six semi-structured interviews were conducted with practitioners to gain insights on developing AI/ML and DS projects using Agile methodologies. A literature review was performed to understand further the challenges and recommendations identified during the interviews. In the second round of interviews, participants assessed the suggested improvements. Qualitative methods were used to analyse the literature review and interview data to address research questions and develop the Agile methodology that caters to AI/ML and DS project-specific requirements and issues.

Results and Discussion: Two primary challenges arise with the data gathered from interviews and literature. Conventional Agile frameworks struggle with the inherent uncertainty, requiring adaptable iterations and flexibility. Effective stakeholder engagement and transparency are also necessary, involving business users and domain experts throughout the Agile process. To address these challenges, a customised Agile methodology is proposed that combines Kanban, Design Thinking, and Lean Startup Life Cycle. By integrating these methodologies, challenges related to task flexibility, uncertainty, and stakeholder communication can be addressed. The proposed approach included the best practices identified for implementing the customised Agile methodology in AI/ML and DS projects, such as collaborative efforts, flexible iterations, specialised roles, prioritised experimentation, and feedback loops.

Conclusion: In summary, this study examines the development and customisation of Agile methodologies for AI/ML and DS projects. The research aims to bridge the gap between Agile practices and the intricacies of AI/ML and DS projects by addressing specific challenges, suggesting a tailored approach, and emphasising effective methods. Future studies could build upon these findings by exploring practical applications and assessing the effectiveness of the proposed methodology in various settings.

Acknowledgement

I want to express my sincere gratitude to everyone who supported me throughout my research and writing of this thesis. Firstly, I want to thank Dr. Marc Hilbert, my first supervisor, for his unwavering guidance, insightful feedback, and constant encouragement. His expertise and dedication significantly shaped the direction of this research and enriched its quality. Similarly, I am equally grateful to my second supervisor, Dr Christoph Stettina, for his valuable insights, constructive criticism and thoughtful guidance that helped refine the scope and methodology of this study.

I am immensely grateful to the interview participants who generously shared their time, expertise, and insights. Their candid feedback and thoughtful perspectives have greatly enriched the depth and relevance of this research. I am also grateful to the thesis support group led by Alexandra Blank and fellow group members who provided unwavering encouragement, thoughtful discussions, and invaluable suggestions throughout this journey. Their camaraderie and support have been a constant source of motivation.

I would also like to thank the academic and non-academic staff at LIACS for their support throughout my master's journey. My appreciation goes to my family and friends for their unwavering support, patience, and encouragement throughout this endeavour. Their belief in me has been a constant source of motivation. Lastly, I am grateful to the institutions, libraries, and resources that provided access to the knowledge and materials essential for completing this thesis.

This work would not have been possible without the collective efforts of all those mentioned above. Thank you for being an integral part of this journey.

Table of Contents

Figures

Tables

1 Introduction

Adaptive technologies such as Artificial Intelligence (AI) and Machine Learning (ML) have revolutionised how businesses operate, allowing them to harness data to extract valuable insights, streamline processes, and make informed decisions (*McKinsey Technology Trends Outlook 2022*, 2022). According to McKinsey's recent survey, the adoption of AI has surged to 56% in 2021 from 50% in the previous year, reflecting a growing trend towards integrating AI in business operations (*McKinsey Technology Trends Outlook 2022*, 2022). However, despite the proliferation of AI applications, organisations must overcome various organisational, technical, ethical, and regulatory challenges to leverage this technology's full potential (*McKinsey Technology Trends Outlook 2022*, 2022).

Agile methodology has proven advantageous to organisations implementing it within and beyond the software development realm. Despite the growing popularity of Agile approaches in software development, AI, ML, Data Science (DS), and research projects often take longer than an agile cycle and come with uncertainties. A study by Vaidhyanathan et al. (2022) mentioned that when they started a new ML project, integrating ML teams into the agile methodology posed challenges due to ML development's experimental and iterative nature, making communication and collaboration difficult. As a result, adopting agile practices becomes challenging (Vaidhyanathan et al., 2022). This can be challenging for companies looking to stay within budget constraints while carrying out these projects. A customised Agile methodology is needed to address the challenges of AI/ML and DS projects. Although various frameworks are available for AI/ML and DS projects, integrating multiple software development approaches can be challenging for organisations.

An empirical investigation by Lwakatare et al. (2019) identified multiple challenges in ML projects involving machine learning models. Developing applications using ML techniques involves building data-driven ML models (Lwakatare et al.)., 2019). Conventionally, various trials are performed before selecting the final ML model when developing ML models (Lwakatare et al., 2019). ML application development involves applying learning algorithms to a data set and evaluating the accuracy and performance of created ML models (Lwakatare et al., 2019). This should happen in rapid iterations; thus, it is unforeseeable to determine when it is possible to achieve the best-performing model. Also, due to data dependency, the first phase of AI/ML projects goes through numerous challenges related to understanding, validation, cleaning, and enriching data (Polyzotis et al., 2017). When addressing these production challenges, unforeseen activities inevitably emerge during the planning. Due to its limited flexibility, enabling ad-hoc tasks to be completed efficiently and effectively becomes unfeasible within agile methodologies such as Scrum. As a result, developing and maintaining AI/ML systems in the real world presents several hurdles. New evidence emphasises the need to consider and expand established software engineering (SE) principles, methodologies, and tools in developing ML systems to solve these problems (Lwakatare et al., 2019).

This study aims to provide a custom Agile methodology for AI/ ML and DS projects. The challenge is establishing what it takes to merge AI, ML, and data science projects with agile application development. The objective is to develop a more effective methodology to accommodate uncharted constraints and requirements of AI, ML, and data science projects, such as data preparation, experimentation, working with hypotheses, and rapid iteration. This allows organisations to adapt to changing technology and customer needs and respond to the field's moving landscape. In addition, adopting a customised Agile methodology for AI/ML and DS projects enables organisations that already use Agile to continue using their existing measurements, KPIs (Key Performance Indicators) and processes with minimal alterations. The proposed method provides a structured approach to balancing agility and consistency in model development, data discovery, and project management, as well as insights into how businesses can effectively adapt to the highly dynamic AI and ML fields. Introducing a tailored Agile methodology for AI and ML projects can make significant academic contributions to multiple areas, including project management, AI and ML, data science, organisation studies, software engineering, and IT management.

The suggested approach differs from existing related work, which often concentrates on finding specific solutions for problems within a particular phase of AI/ML and DS applications or a specific type of project. Instead, this research focuses on structuring the development process, excluding deployment, monitoring, and control. The developed concept is versatile and can be applied to any AI/ML technology project, such as Data Science, Deep Learning, Big Data, Natural Language Processing, and Image Analysis.

The thesis has a structured format with multiple chapters. Chapter two covers the theoretical background, while chapter three explains the research methodology. Chapter four identifies requirements and problems through interviews and a literature review. Chapter five details the proposed solution's design, development, and evaluation. Chapter six summarises the conceptual solution, and chapter seven discusses findings, limitations, and future research opportunities.

1.1 Research Questions

AI/ML and Data Science fields are constantly evolving, yet managing the development process within these disciplines remains challenging. While agile methodologies have proven effective in software development, their application within AI/ML and DS context still needs to be well-established. Thus, there is a pressing need for a unique methodology tailored to these fields' specific needs. This thesis addresses this issue by exploring the key research questions below.

1. What are the unique challenges and requirements of AI/ML and DS projects that are not addressed by existing agile methodologies?
2. How can agile methodologies be customised better to suit the needs of AI/ML and DS projects?

3. What are the best practices for implementing a customised agile methodology in AI/ML and DS projects?

2 Literature Review

2.1 Introduction

AI and ML fields rapidly expand, offering numerous applications in various industries. However, the unique characteristics and complexities of AI and ML projects have presented challenges in managing and delivering these projects (Lwakatare et al.)., 2019). As a result, most machine learning projects fail to reach production, with an 87% failure rate (Ranawana & Karunananda, 2021). Building a functional (AI/ML) prototype can take at least six months due to deployment, scaling, and versioning challenges (Ranawana & Karunananda, 2021). A study conducted by Westenberger et al. (2022) identified twelve factors categorised into five that lead to the failure of AI projects: unrealistic expectations, use case-related issues, organisational constraints, lack of key resources, and technological issues. Idealistic expectations about AI's capabilities can arise, leading to products labelled as AI failing to gain adoption (Westenberger et al., 2022). Westenberger et al. (2022) also highlight that the success of AI initiatives can be hindered by use case issues, such as the complexity of a use case surpassing internal development teams' abilities. Use cases, such as autonomous transportation, require accurate and precise predictions and outcomes, as errors can have fatal consequences (Westenberger et al., 2022). Further, AI projects can be risky because their outcome is hard to anticipate, resulting in insufficient resources being allocated and early termination due to budget constraints (Westenberger et al., 2022). Furthermore, organisational and resource constraints such as failing to hire experts, acquiring training data, and not conducting training due to resources and budget overruns could make the projects fail (Westenberger et al., 2022).

2.2 Theoretical Background

2.2.1 AI, ML and Data Science

Artificial Intelligence and Machine Learning are not new concepts (Alzubi et al., 2018). They have been extensively studied, implemented, and evolved by computer scientists, engineers, researchers, students, and industry professionals for over 60 years (Alzubi et al., 2018). For the past four decades, companies have developed their analytical capabilities using statistics and other quantitative methods to aid decision-making. They are increasingly interested in exploring and utilising AI. (Davenport, 2018). AI, ML, and Data Science are computer science disciplines that are rapidly growing.

John McCarthy, credited with coining the phrase "Artificial Intelligence" in 1956, conceptualised that AI was applying science and engineering to develop intelligent machines for human use (Rupali & Amit, 2017). AI is a modern technique for using devices to carry out strenuous activities and solve challenging issues using intellectual capabilities (Rupali & Amit, 2017). AI encompasses various areas, including computer science, mathematics, psychology, linguistics, philosophy, neuroscience, and artificial psychology (Ongsulee, 2017). Artificial intelligence heavily relies on machine learning, essentially a statistical and analytical approach (Davenport, 2018).

Machine learning (ML) is a subset of AI that allows computers to think and learn independently at a basic level (Alzubi et al., 2018). In 1959, Arthur Samuel introduced "Machine Learning" as the computer learning process without requiring explicit programming (Alzubi et al., 2018). Machine learning is a branch of computer science that originated from studying pattern recognition and computational learning theory in artificial intelligence (Ongsulee, 2017). Many AI methods rely on machine learning, essentially a statistical approach (Davenport, 2018). It has been used for several decades and is commonly known as "predictive analytics" (Davenport, 2018). ML models involve creating a statistical model based on known data values for the outcome variable (Davenport, 2018). Machine learning also includes sophisticated model types, such as neural networks and deep learning, which rely on statistics (Davenport, 2018). Hayashi and Chikio define data science as a paradigm that combines statistics, data analysis, machine learning, and similar techniques to examine real-world phenomena using data (Alzubi et al., 2018).

Integrating AI/ML into smart apps can revolutionise work, learning, and entertainment, transforming industries like banking, healthcare, and transportation (Alzubi et al., 2018). AI/ML capabilities include voice/image recognition, fraud detection, and predicting traffic patterns (Alzubi et al., 2018).

2.2.2 AI, ML and Data Science Workflow

In machine learning, six key components are universal regardless of the technicalities (Alzubi et al., 2018). These include data collection and preparation, feature selection, algorithm choice, model and parameter selection, training, and performance evaluation (Alzubi et al., 2018). Even though different organisations have distinct workflows, AI/ML workflow shares similarities with workflows specified in the context of data science and data mining, such as TDSP (Team Data Science Process, KDD (Knowledge Discovery in Databases), and CRISP-DM (Cross-Industry Standard Process for Data Mining) (Amershi et al., 2019). Amidst having modest changes, these structures share the data-centric nature of the process and the several feedback loops between the various stages. Figure 1 illustrates a simplified view of an AI/M workflow (Amershi et al., 2019). The non-deterministic nature of machine learning systems makes it challenging to construct them using sequential development techniques.

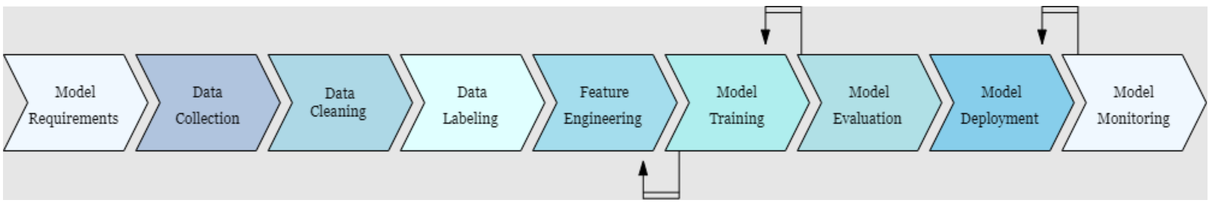

Figure 1 - AI/ML/Data Science Workflow (Ranawana & Karunananda, 2021)

In an ML project, the first step is to train the algorithm by identifying features or patterns in data. This model is then used to make predictions or decisions based on testing data (M. Uysal, 2022). As the algorithm's performance improves, its predictions or decisions become more accurate (M. Uysal, 2022).

Thus, a model is created by combining data with an appropriate algorithm that must fit into the solution space (M. Uysal, 2022). Machine learning models are typically classified into three categories: supervised learning (SL), unsupervised learning (UL), and reinforcement learning (RL) (M. Uysal, 2022). Here are the primary stages of a machine learning project:

i. Problem definition
ii. Data collection
iii. Data cleaning/processing
iv. Feature extraction
v. Model training
vi. Model evaluation
vii. Deployment and monitoring

ML projects share similarities with DS projects in problem definition, data acquisition, and data processing stages (M. Uysal, 2022). Additionally, ML is connected to the modelling stages of DS projects (M. Uysal, 2022). The following steps will be discussed below: collecting and preparing data, extracting features, training and testing the model, evaluating the model, deploying the model, and monitoring it.

I. Data collection and preparation

   Collecting and preparing data is the primary and most difficult step in developing an ML model. Data quality, structure, and relevance play a crucial role in a machine learning model's performance, and various sources can provide data of varying levels of quality and format (Ranawana & Karunananda, 2021). The data must be gathered, cleaned, augmented, and labelled for use in machine learning model development, and it takes time for engineers to understand the available data (Ranawana & Karunananda, 2021). This understanding is necessary for successful data engineering and model debugging (Ranawana & Karunananda, 2021). Consistency in data is also essential, particularly for systems that require large amounts of manually labelled data (Ranawana & Karunananda, 2021). To reduce the impact of data collection on model development, data collection and labelling should be done simultaneously with model development and training (Ranawana & Karunananda, 2021). The standard production strategy begins with sufficient data (Ranawana & Karunananda, 2021). This allows teams to construct the modelling pipeline and evaluation infrastructure without waiting for data collection to be completed (Ranawana & Karunananda, 2021). ML processes are based on data science and data mining workflows and, as such, reflect the data dependency of machine learning model development (Ranawana & Karunananda, 2021).

II. Feature Engineering (Extraction)

After acquiring and processing the data, the next stage is Feature Engineering (FE). This involves selecting the most appropriate features from the raw data and making them available for the Modelling, Training, and Testing stages of ML model development (Uysal, 2022). FE transforms unstructured and raw data components into suitable data formats for learning algorithms (Uysal, 2022). Features are intermediaries between data and models, so they must be derived carefully. Filtering, wrapping, and embedding methods are used for feature selection, and sometimes modelling techniques are used to derive features (Uysal, 2022). However, FE methods performed before model training and testing can have adverse effects, such as overfitting and low performance (Uysal, 2022). In addition, the FE and data processing stages can be complicated and time-consuming, taking up most of the time and resources (Uysal, 2022).

III. When solving a research problem, machine learning models use algorithms and data to find solutions efficiently and accurately (Uysal, 2022). The ideal model should be precise, interpretable, and maintainable and applicable (Uysal, 2022). To create a model, algorithms are used in a training process where the model is fed with data to build a mathematical representation between the data and the target (Uysal, 2022). Then, learning algorithms look for patterns in the data to map the input to the target, allowing them to make predictions or decisions without explicit programming (Uysal, 2022).

ML modelling involves using different techniques to help computers learn and make predictions or decisions based on data. There are four main learning methods in machine learning: supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. Each method has distinct features and applications, which provide useful tools for solving various problems in AI/ML. Therefore, machine learning engineers can apply the appropriate techniques to solve various problems by understanding and utilising these different learning methods.

a. Supervised Learning

Supervised Learning (SL) is a popular machine learning technique where a model learns to make predictions from labelled training data (Uysal, 2022). The training data contains example inputs and corresponding labelled outputs, which the model uses to map new testing or unseen data to new results (Uysal, 2022). SL algorithms are typically used for classification and regression problems (Uysal, 2022). Major SL algorithms include k-Nearest Neighbor (k-NN), Linear Regression, Logistic Regression, Support-Vector Machines, Decision Trees, Random Forest, Naive Bayes, and Neural Networks. Ensembles are methods that combine multiple ML algorithms to create more powerful learning models (Uysal, 2022).

b. Unsupervised Learning

Unsupervised Learning (UL) is a model training technique that analyses data without labels to discover patterns and clusters (Uysal, 2022). UL algorithms aid in Exploratory Data Analysis (EDA) and may include dimensionality reduction tasks to represent the data's essential characteristics (Uysal, 2022). In addition, UL models are used for preprocessing in Supervised Learning (SL) models to enhance their accuracy and performance (Uysal, 2022). However, scaling extensive training data and comparing outcomes between different models can be challenging (Uysal, 2022). Major UL algorithms include k-Means, k-Medoids, DBSCAN, Principal Component Analysis, Hierarchical Clustering, and Hidden Markov (Uysal, 2022).

c. Reinforcement Learning

Reinforcement Learning (RL) is a trial-and-error approach to learning through interaction with a changing environment (Uysal, 2022). The method uses labelled feedback to change its status as it navigates the environment (Uysal, 2022). RL models are helpful when examples of desired behaviours are unavailable, and behaviours can be scored based on criteria (Uysal, 2022). In ML, reinforcement learning algorithms utilise dynamic programming techniques (Uysal, 2022). These algorithms can represent their environments using finite Markov decision processes and Monte Carlo methods (Uysal, 2022).

IV. Model Testing, Validation and Evaluation

When working with ML models, it is essential to train them and validate and select the most suitable model (Uysal, 2022). Model validation can be done using quantitative and qualitative methods (Uysal, 2022). Quantitative methods focus on the data and models, while qualitative techniques consider the ML process life cycle. Metrics for evaluating different types of models, such as classification, regression, and clustering, vary (Uysal, 2022). For example, classification tasks use metrics like accuracy, precision-recall, confusion matrix, the area under the curve, and log-loss, while regression tasks use the root-mean-square error (Uysal, 2022). In addition, cross-validation and hold-out strategies are used for testing and validation (Uysal, 2022). A good ML model should have good predictive power and generalise well to unseen data (Uysal, 2022). Common modelling errors for classification tasks include overfitting and underfitting, which occur when the model is too complex or too simple, respectively (Uysal, 2022).

V. Deployment and Monitoring

The final stage of a machine learning project is the most tangible for users, customers, and stakeholders (Uysal, 2022). It can involve producing reports, developing complex software applications, and integrating the model with various systems (Uysal, 2022). Monitoring and measuring the benefits of the deployed solution is also essential (Uysal, 2022). In addition, a detailed plan can help with day-to-day business actions and maintenance activities (Uysal, 2022).

A. Machine Learning and Data Science Team

Machine Learning and Data Science teams handle complex projects with systems, software, and data engineering (Uysal, 2022). Different roles are assigned depending on the project management approach, but the common ones are related to system engineering, software engineering, and data/ML engineering (Uysal, 2022). Business analysts, system analysts, data scientists, and data/ML engineers are all involved in these projects (Uysal, 2022). Business analysts provide business expertise, while system analysts define and design system and software integration documents (Uysal, 2022). Data engineers handle data preparation and processing (Uysal, 2022). Data scientists perform statistical analysis and develop DS models (Uysal, 2022). Some projects may involve software architects and backend/frontend software developers. Roles may vary depending on the organisation's practices (Uysal, 2022).

2.2.3 Software Development Life Cycle

The Software Development Life Cycle (SDLC) refers to the activities involved in creating or maintaining software systems (Leau et al., 2012). The life cycle includes stages from requirement analysis to post-production testing and evaluation (Leau et al., 2012). In addition, it consists of the models and methodologies used by development teams to build these software systems, which provide a structure for organising and managing the entire development process (Leau et al., 2012). The history of software development began in the 1950s, but early approaches were not effective for larger and more complex software products (Ranawana & Karunananda, 2021). The Waterfall model was developed in 1970, followed by iterative software development and the spiral model (Ranawana & Karunananda, 2021). Agile methodologies were introduced because of the need for rapid feedback-based software development, allowing flexibility and adaptability to change through continuous planning, improvement, development, and system deployment (Ranawana & Karunananda, 2021). The Agile SDLC is based on the unified process model, as shown in Figure 2 (Ranawana & Karunananda, 2021). These steps are usually repeated in short iterations, following an iterative model. Figure 2 shows the phases in a typical software development life cycle (Ranawana & Karunananda, 2021).

Figure 2- software development life cycle (Ranawana & Karunananda, 2021)

2.2.4 Software Development and AI, ML, & Data Science Application Development

Due to their fundamental distinctions, software engineering and machine learning cannot be treated uniformly with the same methodologies (Ranawana & Karunananda, 2021). While software engineering relies on software design, development, and testing, machine learning model development involves data and model design, training, evaluation, deployment, and monitoring (Ranawana &

Karunananda, 2021). Singla et al. (2018) conducted a study that found substantial differences in Agile software engineering project execution between a machine learning team and a non-machine learning team. Singla et al. (2018) point out that ML teams usually develop software modules for internal use in other applications, while non-ML teams cater to end-users. These projects involve applying various machine learning algorithms to data for predictions, and the team may need to repeat experiments with different approaches (Singla et al., 2018). As a result, predicting the duration of machine learning projects can be challenging, as the outcomes are not always foreseeable (Singla et al., 2018).

Software system behaviours are initially specified and defined using various design models and programming codes (Uysal, 2022). In contrast, machine learning systems continuously learn system behaviours after processing training and testing data sets (Uysal, 2022). Minor modifications to the system input can drastically alter system behaviour, needing specialised testing, validation, and verification techniques for ML initiatives (Uysal, 2022). Analysis and specification of detailed and exhaustive requirements may only be possible for ML after a while (Uysal, 2022). Moreover, the black-box nature of ML algorithms makes it challenging to explain "what is possible and what is not" to both technical and non-technical stakeholders (Ishikawa and Yoshioka, 2019).

According to a recent study by Wan et al. (2021), there are significant differences in software engineering practices between ML and non-ML development. Specifically, ML development requires more preliminary experiments to collect requirements, which can lead to predictable degradation in performance (Wan et al., 2021). ML systems have uncertain requirements as they aim to improve decision-making rather than provide functional ability (Wan et al., 2021). Requirements include a conceptual goal after applying the system, which can vary based on the data and context of the application (Wan et al., 2021). Additionally, the detailed design of ML systems is more time-consuming and tends to be conducted iteratively (Wan et al., 2021). Furthermore, when it comes to testing and quality, collecting a testing dataset requires more effort in ML development, and good performance during testing does not guarantee success in production (Wan et al., 2021). Finally, the availability of data can limit the capability of ML systems, making data processing a critical factor in the overall success of the process (Wan et al., 2021).

Amershi et al. (2019) also highlighted some key distinctions between designing applications and platforms for training and deploying AI and ML models versus conventional software development. The primary technological hurdles of AI/ML development include data, hidden technical debt, and the necessity for iterative experimentation (Ranawana & Karunananda, 2021). ML projects depend entirely on data; the effort and precision required to maintain and version data are intrinsically more complex than software code (Amershi et al., 2019). Applications that use ML/AI are called data-driven as they are developed primarily based on existing data, recorded experiences, or simulations rather than solely relying on rule-based knowledge like traditional information systems (Hesenius et al., 2019). Due to its dependency on data, many machine learning projects cannot predict the accuracy of a classification task, regression, or recommendation's relevance before implementation (Singla et al., 2018).

To design, assess, and modify models from scratch, the customisability and extensibility of models demand both software engineering abilities and in-depth knowledge of machine learning (Amershi et al., 2019). For data-driven applications, new skills are needed to manage and analyse data (Franková et al., 2016). This has led to new roles in ML/data science teams, like data scientists and data/ML engineers, each requiring unique skill sets (Franková et al., 2016). Although the roles of data engineer and software engineer may appear similar, the skills and knowledge necessary to become a data engineer significantly differ from those required to become a software engineer (Krasteva & Ilieva, 2020). For example, data scientists need business knowledge, machine learning, data analysis, mathematics, operations research, programming, and statistics. Such roles require different skills than software engineers (Franková et al., 2016).

Managing and versioning data during the development process, monitoring and logging data for implemented models, and assessing the effort required to build ML components present notable distinctions from developing conventional software components (Serban et al., 2020). Due to the lack of a test oracle, their frequently non-deterministic behaviour, and the complexity of defining test coverage, it is notably difficult to test and ensure the quality of ML components (Serban et al., 2020).

Amershi et al. (2019) also mentioned that due to intricate interdependencies between models during training and tuning, it is more difficult to enforce precise module boundaries between machine learning components than between software engineering modules.

Kelly and Kaskade (2013) suggest that ML and data science projects focus on the timeliness, quality, and availability of data, unlike other categories of SE and information system projects (Uysal, 2022). In addition, the rapid expansion of data demands adding computing and storage resources to data processing environments (Larson & Chang, 2016). In a study conducted by Clemmedsson (2018), it was found that the most common technical challenges faced in ML projects are inadequate and inappropriate data, incorrect data forms and feature selection, issues with model or algorithm usage, biased model training, overfitting, and inadequate testing. In addition to advances in technology and extensive data resources, data storage and administration pose significant threats to the overall success of a project (Uysal, 2022). Adaptability, scalability, safety, and privacy are additional challenges for developing large-scale ML systems in industrial contexts (Lwakatare et al., 2020).

2.2.5 Agile Methodologies

The manifesto and principles of Agile Software Development (ASD) were published in 2001 (Larson & Chang, 2016). These principles have been incorporated and evolved to develop various Agile approaches (Larson & Chang, 2016). Agile methodologies are a set of principles for software development that emphasises flexibility, rapid iteration, and collaboration between teams (Schön et al., 2015). The practices of self-organizing teams that work at their own pace to sustain creativity and productivity are at the core of agile development (Dingsøyr et al., 2012). These practices allow for changes in requirements at any stage of development and involve customers or their representatives in the process for feedback and reflection (Dingsøyr et al., 2012). While not a formal definition of agility, these principles serve as guidelines for delivering high-quality software in an agile manner (Dingsøyr et al., 2012).

Agile methodologies have been widely adopted in software development and effectively manage and deliver software projects. Compared to other project or software development approaches, agile methodologies have been found to achieve success by attaining lower cycle times, improved quality, better requirement clarity, increased flexibility, and higher satisfaction levels among stakeholders (Larson & Chang, 2016).

There are various popular agile methods such as Scrum, Kanban, Extreme Programming (XP), lean software development, Dynamic Systems Development Method (DSDM), Crystal methods, Test Driven Development (TDD), Feature Driven Development, and Scaled Agile. Although they may have different activities and objectives, they share the same core values. Agile frameworks like Scrum, Kanban, and XP provide a structured approach to implementing Agile principles (Schön et al., 2015).

Software development using Agile methods involves different stages in the development life cycle (Abrahamsson et al., 2010). However, since various practitioners developed these methods independently, there is no clear explanation for their focus on specific aspects (Abrahamsson et al., 2010). This makes it challenging to choose the most appropriate way without a thorough understanding of their coverage in the life cycle (Abrahamsson et al., 2010). Additionally, the lack of clear explanations makes it more challenging to determine how the covered and uncovered life cycle phases connect, making applying these methods more complicated (Abrahamsson et al., 2010).

I. Scrum

Scrum is the most widely used agile method. The Scrum methodology, created by Schwaber in 1995 and later improved by Beedle in 2002, is specifically tailored to manage software development in dynamic environments. It prioritizes flexibility, adaptability, and productivity (Abrahamsson et al., 2010). The Scrum methodology allows developers to choose the software development techniques, methods, and practices for implementation. It involves regular management activities to identify deficiencies or impediments in the development process and the approaches used (Abrahamsson et al., 2010).

In Scrum, the software product is delivered in a series of iterations or increments within a predefined time box called sprint, which is, at most, four weeks (Sharma & Hasteer, 2016). A shippable product increment is delivered to the user at the end of each sprint (Hron & Obwegeser, 2018). A Scrum team comprises three key roles: a product owner, a Scrum master, and a development team (Sharma

& Hasteer, 2016). The developers and other stakeholders select tasks for the sprint together during a planning meeting before each new sprint. The product owner represents the customer, and requirements are captured as user stories in a prioritised product backlog that is continuously updated (Hron & Obwegeser, 2018). The Scrum master is crucial in leading and guiding the development team, resolving issues, making necessary improvements, and acting as a communication bridge between the product owner and the development team (Sharma & Hasteer, 2016). The Scrum Master is responsible for leading the Daily Stand-up meetings (typically 15 minutes), where team members update each other on their progress and tasks for the day to keep work moving quickly (Hron & Obwegeser, 2018).

The Scrum process begins with the product owner's vision for the product they want to create (Sharma & Hasteer, 2016). Next, they make a prioritised list of product features called the product backlog (Sharma & Hasteer, 2016). A sprint is initiated with sprint planning and a sprint backlog, including the tasks to be completed by the team (Sharma & Hasteer, 2016). During sprint planning, the team selects tasks from the product backlog they can complete within a sprint cycle (Sharma & Hasteer, 2016). In sprint planning, the team breaks down tasks and strategies to achieve goals within each sprint cycle (Sharma & Hasteer, 2016). At the end of each sprint cycle, a sprint review is conducted, where all stakeholders, including customers, the Scrum team, and associated members, inspect the product and provide feedback (Sharma & Hasteer, 2016). Finally, before the next sprint planning, the scrum team conducts a sprint retrospective to identify areas for improvement and achieve better results in the upcoming sprint phase (Sharma & Hasteer, 2016).

Scrum methodology breaks down products into manageable chunks, allowing progress even with varying requirements (Rising & Janoff, 2000). This leads to better communication, shared successes, on-time delivery, and frequent feedback for customers (Rising & Janoff, 2000). In addition, trust between the team and the customer is fostered, leading to a culture of success (Rising & Janoff, 2000). Akif and Majeed (2012) found that Scrum implementation can be challenging due to management, development, and release process issues. The obligation of delivering a shippable product at the end of each sprint may cause teams to overlook software quality, and working with multiple teams can take time and effort (Akif & Majeed, 2012). While Scrum has a "Scrum of Scrums" technique, it may not work well in a distributed environment (Akif & Majeed, 2012). Although teams are expected to self-manage, this may not always be the case, particularly with new team members (Akif & Majeed, 2012). In addition, communication in Scrum can be overwhelming, leading to frequent meetings that may only be relevant to some team members (Akif & Majeed, 2012).

II. Kanban

Kanban has gained popularity in software development over the past few years. The Lean and Kanban approaches were first implemented in the Japanese manufacturing industry in the 1950s (Ahmad et al., 2018). Kanban, a Japanese term for "signboard," was used as a scheduling system in manufacturing. In 2004, David J. Anderson introduced the Kanban method in software development while assisting a struggling small IT team at Microsoft (Ahmad et al., 2018). This method encourages project teams to visualize their workflow, limit work in progress (WIP) at each stage, and measure cycle time (Ahmad et al., 2018). Using a Kanban board makes the software development process visible, providing clear communication of priorities, highlighting bottlenecks, and showing the assigned work of each developer (Ahmad et al., 2018). The goal of the Kanban method is to minimize WIP by focusing only on requested items, resulting in a constant flow of released work items to customers (Ahmad et al., 2018). This method also aims to quickly adapt the process by using shorter feedback loops and focusing on flow without mandatory iterations (Ahmad et al., 2018).

Implementing Kanban begins with current processes, roles, responsibilities, and job titles and gradually improves. This approach fosters leadership at all levels and reduces cycle time (Flora & Chande, 2014). Assess the roles that work best for the team and determine whether to add or remove roles based on their impact on the process (Flora & Chande, 2014).

Despite experiencing benefits, some challenges were encountered while using Kanban (Kirovska & Saso, 2015). The most prevalent challenge was a need for more familiarity with Kanban, which led to difficulties in managing WIP limits and task prioritisation (Kirovska & Saso, 2015). Another obstacle was the organisation's traditional culture and its impact on adopting Kanban (Kirovska & Saso, 2015).

III. Extreme Programming

In 1998 Kent Beck, Ron Jeffries, and Ward Cunnigham introduced Extreme Programming (XP) (Flora & Chande, 2014). The XP methodology allows small teams of developers to create software efficiently and iteratively, resulting in better quality and productivity (Matharu et al., 2015). In addition, it involves close collaboration with customers throughout development to adapt to changing requirements (Matharu et al., 2015). The process consists of collecting informal customer requirements on-site, forming pairs of programmers, creating basic designs, and continuously refining, integrating, and testing (Flora & Chande, 2014). It also promotes frequent releases in short development cycles, which increases productivity and provides opportunities to incorporate new customer requirements (Flora & Chande, 2014).

A team of 2 to 12 members working in the same location is recommended for optimal results. Iterations typically last 1 to 3 weeks and have 6 phases in the XP methodology (Flora & Chande, 2014). These phases include Exploration (writing stories for the current iteration), Iteration

Planning (prioritising stories and estimating effort and resources), Iteration to Release (analysis, design, coding, and testing), Production (rigorous testing), Maintenance (providing customer support and releasing for use), and the Death Phase (when no further requirements exist) (Flora & Chande, 2014). In the XP process, customers represent the requirements as scenarios, which are then transformed into Story Cards (Matharu et al., 2015). Next, the developers break down each Story Card into small tasks (Matharu et al., 2015). Finally, the customer prioritises these tasks for implementation (Matharu et al., 2015).

XP has limitations and may not be suitable for large, complicated, or complex projects (Flora & Chande, 2014). Coordination between programmers during pair programming is crucial, and any conflicts could negatively impact the collective code ownership objective (Flora & Chande, 2014). Pair programming is vital in XP but can only be applied to projects with multiple developers (Flora & Chande, 2014). Also, customer collaboration may be weak, and testing and code development are often done by the same person, which may result in some problems being overlooked due to the developer's perspective (Flora & Chande, 2014).

Different Agile methodologies may adhere to the same principles of the Agile Manifesto, but they have variations in various aspects. Table 1 compares Scrum, XP, and Kanban based on the studies by Matharu et al. (2015) and Dingsøyr et al. (2012).

Parameters	Scrum based Development	Extreme Programming (XP)	Kanban Methodology
Design Principle	Complex design	Simplification of code and accommodation of unexpected changes through refactoring	Limits the amount of work-in-progress and ensures waste reduction
Nature of Customer Interaction	Not compulsorily on-site	On-site Customer Interaction	Not compulsorily on-site
Design Complexity	Complex design	Simple design	Simple visual design
Project Coordinator	Scrum Master	Teamwork	Teamwork
Roles Assigned	3 pre-defined roles: Product Owner, Scrum Master & Development Team	No prescribed roles	No prescribed roles
Process Ownership	Scrum Master	Team ownership	Team ownership
Product Ownership	Product Owner is responsible for product	Not prescribed	Not prescribed
Team Collaboration	Cross functional teams	Self-organising teams	Self-organising teams
Workflow Approach	Sprints	Short iterations	Short iterations
Requirements Management	Requirements managed in form of artifacts through sprint backlog and product backlog	Managed in form of story cards	Managed using Kanban boards
Product Delivery	Delivery as per time boxed sprints	Continuous delivery	Continuous delivery
Coding Standards	No coding standards	Coding standards are used	No coding standards
Testing Approach	No formal approach used for testing	Test-driven development, including acceptance testing	Testing done after implementation of each work product
Accommodation of Changes	Changes not allowed in sprints	Amenable to change even in later stages of development	Changes allowed at any time

Table 1 - Comparison among Scrum, XP and Kanban Agile Methodologies Kanban (Matharu et al., 2015)

IV. Scaled Agile Framework (SAFe)

Dean Leffingwell created the Scaled Agile Framework in 2011 to help large enterprises adopt agile practices (Putta et al., 2018). The framework combines Scrum, Extreme Programming, Kanban, and Lean methods and is divided into four levels: Team, Program, Portfolio, and Value Stream (Putta et al., 2018). At the Team level, agile teams work together (Putta et al., 2018). The Program level introduces Agile Release Trains (ARTs) to coordinate large groups of teams and individuals (Putta et al., 2018). ARTs use HIP (Hardening, Innovation, Planning) iterations to create Potential

Shippable Increments (PSIs) or Program Increments (PIs), which are planned during release planning days (Putta et al., 2018).

In SAFe, Agile teams operate like Scrum teams with a few differences (Alqudah & Razali, 2016). On the Team level, SAFe adopts Scrum, which includes a Scrum Master, a Product Owner, and 5-9 members who work together to create cohesive end-user value (Alqudah & Razali, 2016). New roles and teams are established at the program level, including the Product Manager, System Architect, Release Train Engineer, and User Experience Designer (Alqudah & Razali, 2016). SAFe also has additional program-level teams and a value stream level for larger products (Alqudah & Razali, 2016). The Program Portfolio Management team provides portfolio vision, funding, and governance (Alqudah & Razali, 2016).

Adopting SAFe can increase transparency, alignment, quality, predictability, and productivity (Putta et al., 2018). According to practitioners, SAFe can provide various business benefits, such as faster time to market and more frequent deliveries (Putta et al., 2018). When adopting SAFe, organisations may face challenges such as resistance to change, moving away from agile, program increments planning, controversies with the framework, Agile Release Train challenges and staffing roles (Putta et al., 2018). Resistance to change can be supported by the 12th State of Agile survey results, which demonstrated general resistance among organisations (Putta et al., 2018). Additionally, some organisations may feel that SAFe moves away from agile principles, as argued by some "Agilists" such as Ken Schwaber, Ron Jeffries, and Stephen Denning (Putta et al., 2018). Furthermore, some organisations need clarity within the framework, such as overhead and story point normalisation (Putta et al., 2018). Moreover, no clear roadmap is available to assist enterprises in preparing for and adopting SAFe (Turetken et al., 2017). While SAFe outlines agile and lean principles, it does not provide specific implementation strategies or methods (Turetken et al., 2017). As a result, companies may struggle to identify priorities and effectively implement SAFe practices (Turetken et al., 2017).

SAFe is currently on version 6, and it remains unclear whether they have addressed existing challenges or if new ones have emerged. However, empirical research on the SAFe framework is lacking due to its recent development.

2.2.6 Agile in AI, ML and Data Science

In general, no single approach works for all projects regarding project execution (Krasteva & Ilieva, 2020). Agile methodologies have been widely used in software development to meet these challenges, but current Agile methodology may not fully meet the specific needs of AI and ML projects (Batra, 2017). Hence, various enterprises have formulated customised methods to design machine learning programs (Ranawana & Karunananda, 2021). According to a study by Serban et al. (2020), the practices generally apply to all ML applications and are unaffected by the kind of data being used (Serban et al., 2020). However, AI/ML projects have distinct traits and complications that Agile frameworks may not adequately address.

An important consideration is that implementing Agile methodology in Data Science can require careful planning and strategic execution, particularly for large projects with complex data models (Jurney, 2013). Due to the extensive research involved, it requires exhaustive effort within an unpredictable timeline (Jurney, 2013). Conducting data science research with real-world data takes considerable time, often months longer than an agile cycle (Jurney, 2013). For many organisations, project sprints need to provide more time for such tasks, leaving researchers and analysts feeling extremely time-constrained (Jurney, 2013). Schleier-Smith researched the viability of implementing the agile methodology in machine learning for a mobile dating application's development and design process (Singla et al., 2018). The study identified some obstacles, such as obtaining training data and extended deployment cycles (Singla et al., 2018).

When following an agile approach, the results should meet specific and clearly defined objectives (Grady et al., 2017). However, because analytics development is exploratory, it is impossible to establish detailed requirements with complete certainty (Grady et al., 2017). The final analytics models ' specifics become evident only after the outcomes align with the organisation's needs (Grady et al.)., 2017).

Some challenges exist when using a sprint-based Data Science/ML modelling framework. Using Scrum methodology for ML and DS projects, accurately estimating what can be achieved within a sprint can be challenging (Uysal, 2022). Fixed sprint durations can lead to unrealistic and unrelated backlog items, which may not align with the project's needs (Uysal, 2022). For example, tasks like exploratory data analysis and model evaluation may require larger or smaller backlog item segments (Uysal, 2022). To address this issue, adjusting sprint lengths that can be modified based on the specific ML or DS experimentation procedures being used is crucial (Uysal, 2022). These challenges make it necessary to carefully consider how to implement a sprint-based framework in a Data Science/ML context (J. Saltz & Suthrland, 2019).

In another study by Hukkelberg and Berntzen (2019), participants explained that working according to a sprint schedule can be difficult for data scientists/ML engineers since they work based on hypotheses, which may not always provide immediate value from a management and team lead perspective (Hukkelberg & Berntzen, 2019). Nevertheless, working in sprints allows data scientists to gain insights into what is not working and enables them to test other methods in the subsequent sprint (Hukkelberg & Berntzen, 2019).

Kanban can present challenges due to insufficient support, comprehension, training, and misunderstandings (J. Saltz & Suthrland, 2019). Kanban's flexibility can be a double-edged sword since it does not define project roles or process specifications, making implementation challenging (J. Saltz

& Suthrland, 2019). While the absence of a defined process enables teams to incorporate Kanban into existing organizational practices, it also means that each team may implement Kanban differently (J. Saltz & Suthrland, 2019). In summary, teams that want to implement Kanban must create their processes and artefacts (J. Saltz & Suthrland, 2019).

The same study by Hukkelberg and Berntzen (2019) found that for data scientists/ML engineers to thrive, it is essential for them to have the opportunity to experiment and explore the data (Hukkelberg & Berntzen, 2019). The ability to test and explore different hypotheses was highlighted as a crucial aspect of their job (Hukkelberg & Berntzen, 2019). If the work environment is too rigid, it can impede their ability to perform effectively (Hukkelberg & Berntzen, 2019). It was also noted that while creativity and freedom are essential, management should provide guidance on the more significant problem the team aims to solve (Hukkelberg & Berntzen, 2019). Furthermore, data scientists require considerable autonomy in their work (Hukkelberg & Berntzen, 2019).

Hukkelberg and Berntzen (2019) also highlight the significance of managing expectations within a team and across an organisation. It underscores the need for dedicated time and effort to address different value perspectives (Hukkelberg & Berntzen, 2019). The review highlights the potential disparity between team leads/managers and data scientists/ML engineers in understanding value (Hukkelberg & Berntzen, 2019). While a data scientist may find value in a machine learning implementation that did not meet expectations due to the knowledge gained and lessons learned, a manager might perceive it as a failure and struggle to identify any value. Consequently, the review suggests that integrating new roles into a team necessitates reflecting on value's diverse aspects and perspectives (Hukkelberg & Berntzen, 2019). Overall, the study emphasises the importance of effectively managing expectations and promoting a shared understanding of value to ensure successful collaboration within the team and across the organisation (Hukkelberg & Berntzen, 2019).

2.3 Existing Frameworks for AI/ML and Data Science Projects

Various software development frameworks, such as CRISP-DM (Cross-Industry Standard Process for Data Mining), KDD (Knowledge Discovery in Databases), SEMMA (Sample, Explore, Modify, Model, and Assess), OSEMN (Obtain, Scrub, Explore, Model, iNterpret), TDSP (Team Data Science Process), Lean Data Science and SAFe for Data Science, are used in AI, ML and Data Science projects.

A. CRISP-DM (Cross-Industry Standard Process for Data Mining)

CRISP-DM is the de facto standard and an industry-agnostic process model for implementing data mining projects (Schröer et al., 2021). The CRISP-DM process model is a widely used framework for data mining that is not specific to any industry (Schröer et al., 2021). It involves six iterative phases, starting with understanding the business needs and ending with deployment (Schröer et al., 2021). The main idea, tasks, and outputs of each phase are summarised in Table 2, based on the CRISP-DM user guide (Schröer et al., 2021).

Phase	Short Description
Business Understanding	Assess the business situation, determine available and required resources, define the data mining goal (e.g., classification), and establish data mining success criteria. Create a project plan.
Data Understanding	Collect data from sources, explore and describe the data, and assess data quality. Perform statistical analysis and attribute determination.
Data Preparation	Select data based on inclusion and exclusion criteria, handle poor data quality through data cleaning, and construct derived attributes based on the chosen model.
Modelling	Select the modelling technique to build the test case and model. Evaluate various data mining techniques based on the business problem and data. Set specific model parameters.
Evaluation	Check results against defined business objectives, interpret results, explain further actions, and review the overall process.
Deployment	Plan deployment, create final reports or software components, and establish monitoring and maintenance procedures.

Table 2 - CRISP-DM process model descriptions (Schröer et al., 2021)

The six phases can be iterated upon with some flexibility between steps (J. Saltz & Suthrland, 2019). Teams can move through the phases based on their needs and return to a previous stage if necessary (J. Saltz & Suthrland, 2019). Milestones can also be defined as needed (J. Saltz & Suthrland, 2019). CRISP-DM can be seen as a data mining version of a waterfall model (J. Saltz & Suthrland, 2019).

The Lean Design Thinking Methodology for Machine Learning and Modern Data Projects (LDTM) authors suggest that CRISP-DM has limitations in managing the requirements of current technologies, such as machine learning algorithms (Schröer et al., 2021). To address this, they have integrated design thinking approaches into CRISP-DM. LDTM is not limited to any specific field but focuses on new technologies (Schröer et al., 2021). The CRISP-DM model is an organizational process that can be used with any technology (Schröer et al., 2021). However, it only encompasses part of the project lifecycle, particularly when it comes to machine-learning approaches (Schröer et al., 2021). The CRISP-DM framework guides processes but needs to define project roles clearly (Uysal, 2022). Additionally, CRISP-DM focuses on the work phases rather than how the team should coordinate during those phases (J. Saltz & Suthrland, 2019).

B. KDD (Knowledge Discovery in Databases)

Knowledge Discovery in Databases is the nontrivial process of discovering valid, innovative, potentially beneficial, and eventually comprehensible patterns in data (Fayyad, 1997). KDD framework, commonly used in data mining and analytics, has been found to have limited success in Agile AI/ML and Data Science projects, as highlighted in several studies. KDD's linear and sequential nature conflicts with the iterative and collaborative approaches promoted by Agile methodologies (Larson & Chang, 2016). KDD has a defined process with clear phases, including data selection, pre-processing, modelling, evaluation, and deployment. However, Agile projects demand flexibility, adaptability, and constant feedback, which may need to fit better within the structured framework of KDD (Batra, 2017). In the context of AI/ML and Data Science projects, KDD assumes precise data and problem requirements from the beginning, which may only sometimes be possible due to the dynamic nature of these projects (Singla et al., 2018). Agile AI/ML and Data Science projects focus on experimentation, exploration, and learning, which often involve constantly changing data and problems. However, KDD's linear and prescriptive approach may need to improve the flexibility required for successful Agile projects (Amershi et al., 2019).

C. SEMMA (Sample, Explore, Modify, Model, and Assess)

The SEMMA methodology was created by the SAS Institute (Azevedo & Santos, 2008). SEMMA stands for Sample, Explore, Modify, Model, and Assess, and it refers to the steps involved in conducting a data mining project (Azevedo & Santos, 2008). Although the SEMMA procedure is independent of the data mining tool, it is related to the SAS Enterprise Miner software (Azevedo & Santos, 2008). It aims to help the user by developing DM applications (Azevedo & Santos, 2008). The SEMMA framework, commonly used in traditional data mining projects, may encounter difficulties when utilised in Agile AI/ML and Data Science projects. Various studies have indicated this. The SEMMA method involves a sequential approach that includes sampling, exploring and analysing data, modifying it, building models, and evaluating the outcomes. However, Agile methodologies emphasise iterative and collaborative methods, which may need to align better with the rigid and sequential nature of SEMMA (Lwakatare et al., 2019). Agile projects require flexibility, adaptability, and continuous feedback, while SEMMA assumes a predefined and linear workflow. Similar to KDD, SEMMA assumes that data and problem requirements are well-defined from the beginning, but this may not always be realistic in the dynamic and evolving nature of AI/ML and Data Science projects. (Najdawi & Shaheen, 2021). Agile projects prioritise experimentation, exploration, and learning, which may involve changing data and problem formulations. Hence, the sequential and predefined nature of SEMMA may hinder the agility necessary for successful Agile AI/ML and Data Science projects (Najdawi & Shaheen, 2021). As with KDD, SEMMA is not a software management framework but a data exploration and modelling method, which is essential when developing AI and ML models.

D. OSEMN (Obtain, Scrub, Explore, Model, iNterpret)

OSEMN is a more recent data science pipeline introduced in 2010 by Mason and Wiggins (Saltz & Suthrland, 2019). The OSEMN pipeline comprises five phases (Obtain, Scrub, Explore, Model, and iNterpret) (Saltz & Suthrland, 2019). It is a term used in data science to describe a procedure for data preparation, modelling, and assessment (Saltz & Suthrland, 2019). From a team process standpoint, OSEMN is similar to CRIPS-DM in that the focus is not on how the team should coordinate the work phases but rather on the work phases themselves (Saltz & Suthrland, 2019). In addition, OSEMN omits the business and data comprehension initial phases of CRISP-DM and the implementation of the ensuing analytics (Saltz & Suthrland, 2019).

B. TDSP (Team Data Science Process)

Microsoft introduced the Team Data Science Process (TDSP), a flexible and iterative approach for developing data analytics and AI applications (TDSP, 2021). TDSP combines software engineering practices with data science methods like Scrum, CRISP-DM, and KDD (Uysal, 2022). The process consists of four main stages: understanding the business, acquiring and comprehending data, modelling (including feature engineering, model training, and evaluation), and deployment (Uysal, 2022). TDSP defines the roles of solution architect, project manager, data engineer, data scientist, application developer, and project lead, similar to a software engineering project (Uysal, 2022). As the relevant literature discusses, TDSP may encounter challenges when applied in Agile AI/ML and Data Science projects. The TDSP methodology is well-documented and offers helpful tools and utilities (Martinez et al., 2021). However, it heavily relies on Microsoft services and policies, limiting its broader use. Ideally, a methodology should be independent of specific tools or technologies (Martinez et al., 2021). As Martinez et al. (2021) highlighted in their study, the methodology should guide techniques and activities within a defined domain using rules, methods, and processes (Martinez et al., 2021). Despite its dependence on Microsoft tools, TDSP offers valuable project, team, and data management processes (Martinez et al., 2021).

2.4 Related Work

Recent research has begun to address the challenges of managing AI and ML projects using Agile methodologies. There are several studies (Larson & Chang, 2016; Batra, 2017; Singla et al., 2018; Amershi et al., 2019; Lwakatare et al., 2019; Najdawi & Shaheen, 2021) found that Agile methodologies can be effective in managing AI/ML and DS projects, but that current Agile framework may need some customisation to cater the specific needs of these types of projects. Several studies were conducted to develop customised frameworks in response to the lack of alignment. A study by Blomster and Koivumäki (2021) shows that the ML development process was extremely agile and iterative, with overlapping stages (2021). However, there was compelling evidence that the gates retained their positions until the following phase could be initiated (Blomster & Koivumäki, 2021). Their study aimed to build a customised framework to manage ML development projects for digital marketing operations in marketing organisations (Blomster & Koivumäki, 2021). Kohl et al. introduce a new Agile framework for Rapid Quality-Driven NLP (Natural Language Processing) application development called STAMP 4 NLP, which aims to merge software engineering principles with data science best practices for efficient and successful NLP application development (2021). However, the primary problem of these studies is that they are confined to a single application domain and are difficult to generalise.

Saltz and Suthrland have introduced a framework called SKI, a new agile framework combining Scrum and Kanban elements (2019). Compared to Scrum, SKI uses capability-focused iterations instead of time-based sprints, which allows teams to execute small, logical iterations of varying duration (Saltz & Suthrland, 2019). Compared to Kanban, SKI provides clear guidance on roles, artefacts, and events, enabling teams to achieve the benefits of Kanban more easily (Saltz & Suthrland, 2019). The pilot of SKI in the data science context showed promising results. Still, the primary drawback of this study was that, for the pilot, SKI was applied to small-scale projects operated mostly by data science students (Saltz & Suthrland, 2019). Therefore, further research is needed to explore its usage in real-world scenarios (Saltz & Suthrland, 2019).

Ranawana and Karunananda (2021) suggest a framework called MLASDLC that simplifies the planning, development, and deployment of machine learning applications using parallel processes for software and machine learning engineering. This approach reduces project and machine learning development risks by constantly integrating, evaluating, and producing applications. The MLASDLC framework combines elements from the software development life cycle (SDLC), development operations (DevOps), and machine learning operations (MLOps) to develop machine learning applications. However, the authors caution that the development of the supporting application must not be prioritised over adequately conceptualising, designing, validating, and testing the machine learning system. The study's small sample size and technical differences between the evaluated projects are also mentioned.

Another study by Vial et al., a study of a successful consulting firm at the forefront of AI practice, found that a successful approach to AI projects involves combining traditional project management, agile, and AI workflow logic (2022). The review also highlights conflicts between the established and emerging AI workflow logic and provides four strategies to help practitioners manage AI projects (Vial et al., 2022). Further, they have highlighted three important future research areas to focus on managing AI projects. The first area is to investigate the tactics used to address competing logics, such as reconciliation, decoupling, coexistence, or elimination, as well as the function of agile logic in bridging the gap between traditional project management and AI-workflow logics (Vial et al., 2022). The second is to examine the new roles that have emerged in AI projects and the additional knowledge and responsibilities required of existing roles, such as project managers, agile product owners, and clients (Vial et al., 2022). The third area is to study the conflicts and logic that develop in various circumstances or from the customer's perspective and the tactics customers use to negotiate AI project disputes (Vial et al., 2022).

2.5 Conclusion

This analysis of existing literature covered the theoretical foundations, practices, and challenges associated with implementing Agile methodologies and other frameworks in the context of AI/ML and data science endeavours.

Existing literature shows a discrepancy between Agile techniques, like Scrum, and the iterative and exploratory methods required for AI/ML and data science projects. The nature of AI/ML and DS projects frequently involves intricate, uncertain, and constantly changing requirements, which makes it hard to fit them within the strict confines of time-boxed iterations. This disparity leads to several challenges, including forecasting the time required for exploration tasks, limited flexibility, and a lack of involvement from business users.

It also highlighted the importance of consistent feedback and stakeholder participation in AI/ML and DS projects. Given these projects' challenging needs, collecting and incorporating feedback should be more flexible and adaptable. While conventional Agile practices are useful, they need to be tailored to suit the iterative and exploratory nature of AI/ML and data science work. For example, the Kanban way of working provides flexibility for the AI/ML and DS projects' needs; however, it does not provide concrete implementation guidelines.

Several software development frameworks are used in AI, ML, and Data Science projects, including CRISP-DM, KDD, SEMMA, OSEMN, TDSP, Lean Data Science, and SAFe for Data Science. However, these frameworks are typically designed to focus on either a specific type of project or a particular technical workflow, such as CRISP-DM and OSEMN. They may not consider the broader implications of team collaboration and project execution. Further, TDSP gives a promising overall methodology; most of its practices are linked to its own Microsoft products.

While some studies (Batra, 2017; Kohl et al., 2021; Ranawana & Karunananda, 2021; J. Saltz & Suthrland, 2019) have explored improving Agile methodologies in managing AI/ML and DS projects, a tailored Agile method still needs to be specifically designed to address the unique challenges and requirements of AI/ML and Data Science projects. Furthermore, these studies have identified that the existing methodologies need to be revised to address all the challenges in these types of projects.

In conclusion, the present literature review has laid the foundation for creating a tailored Agile methodology that addresses the specific needs of AI/ML and DS projects. These challenges, such as a lack of flexibility and the need for continuous stakeholder involvement, call for an adaptable Agile approach. By examining previous research, this study paves the way for an improved Agile working method that recognises the complexities of AI/ML and data science and leverages them to enhance project outcomes. The forthcoming chapters will delve into the development of this customised Agile methodology and its practical application, providing a comprehensive solution to the challenges that lie ahead.

3 Methodology

This chapter will discuss the methodological framework used during this research. The Design Science Research Methodology was used to develop a custom Agile methodology for AI/ML and DS projects. The primary goal is to address the challenges and limitations of using conventional Agile methodologies in the context of AI/ML and DS projects.

This research comprehensively explores existing literature and semi-structured interviews with practitioners about their project experiences. By identifying the shortcomings and areas for improvement in Agile practices for AI/ML and DS projects, the aim is to lay the groundwork for a customised Agile methodology that aligns seamlessly with the specific needs of AI/ML and DS projects.

3.1 Design Science Research

As Simon (1996) conceptualised, Design Science Research (DSR) focuses on creating innovative artefacts to solve real-world problems (Hevner & Chatterjee, 2010). The application of this concept is crucial and extensively used in the areas of Information Systems (IS) and Software Engineering. The primary focus is on creating tools to tackle intricate issues and enhance the efficiency of technology-based systems and procedures. (Hevner & Chatterjee, 2010).

At its core, DSR aims to create and assess various artefacts to generate knowledge. These artefacts include models, algorithms, frameworks, design principles, and methodologies (Brocke et al., 2020). Unlike classical research, which mainly observes and explains phenomena, DSR focuses on building practical solutions to specific problems (Brocke et al., 2020). This study aims to develop a tailored agile methodology for AI/ML and DS projects.

Based on its alignment with design science, the fundamental principles and key concepts that underlie DSR and the activities conducted in this research can be summarised as follows:

Based on its alignment with design science, the fundamental principles and key concepts that underlie DSR and the activities conducted in this research can be summarised as follows:

1. **Identify the problem**: Recognise the unique obstacles and limitations associated with AI/ML and Data science projects, such as managing data dependence challenges, overseeing model dependencies, working with exploratory data, and navigating unpredictable timelines. An extensive literature and interview data analysis has been conducted to ascertain how the current Agile frameworks fall short in fulfilling the demands of AI/ML and Data Science project delivery.
2. **Design the solution**: Suggested a customised Agile methodology for AI and ML projects based on the challenge identified in the problem identification stage. This includes incorporating Lean Start-up and Design Thinking principles along with Kanban and modifying existing Agile techniques to better correspond with the distinctive characteristics of AI and ML projects.

3. **Present the solution:** The tailored Agile methodology was presented to the practitioners before the interview round two for their feedback.
4. **Evaluate the solution:** Assessed the soundness of the customised Agile framework in managing the AI/ML project and seek input from practitioners. This feedback was used to develop and enhance the methodology's effectiveness.
5. **Communication:** The final tailored methodology will be shared with research participants to enhance their understanding and manage AI/ML and DS projects using agile methods.

The Design Science Research (DSR) methodology was chosen for this study due to its suitability and alignment with the research problem. DSR emphasises problem-solving and practical relevance, making it ideal for developing a custom Agile methodology for AI/ML and DS projects. Its iterative design cycles allow for continuous improvement, adapting to the dynamic nature of these projects. DSR emphasises understanding stakeholders' challenges and contributes to practical artefacts and theoretical knowledge. This research process involved rigorous problem identification to ensure the resulting methodology addresses the specific needs of the AI and ML community.

3.2 Research Process

3.2.1 Interviews

During the first phase, six semi-structured interviews were conducted with industry experts in these fields to gather insightful information. The semi-structured interview approach allowed us to adjust the questions while maintaining a general structure, ensuring consistent data collection across all participants. The interviews were designed to explore the participants' experiences, perspectives, and insights regarding the challenges, requirements, and potential improvements in managing AI/ML and Data Science projects using Agile methodologies.

During the interviews, open-ended questions encouraged participants to share their experiences and provide detailed descriptions. For example, the participants were asked about specific challenges encountered when applying Agile methodologies in AI/ML and Data Science projects and the essential requirements or considerations in developing an Agile framework tailored for these projects. Also asked how they overcome the existing challenges or success of the communication and collaboration methods they use when managing AI/ML and Data Science projects using Agile methodologies and potential areas for improvement or customisation in existing Agile frameworks to support these projects better. The interview questions are in Appendix 1.

In addition, probing questions were used to delve deeper into participants' responses and comprehensively understand their perspectives. For instance, the participants were asked for specific examples or cases where customisation in Agile frameworks for AI/ML and Data Science projects was needed and the challenges or complexities that arise when managing data-intensive or iterative aspects of these projects within an Agile framework. It also explored potential trade-offs or considerations that must be balanced when combining Agile methodologies with the rigorous requirements of AI/ML and Data Science projects.

The first round of interviews aimed to gather comprehensive insights into the challenges and customisation requirements in managing AI/ML and Data Science projects using Agile methodologies. These insights served as a foundation for developing a customised agile framework that aligns with the specific needs and characteristics of AI/ML and Data Science projects.

3.2.2 Data Collection

For this study, a cohort of six distinguished industry experts was selected with extensive knowledge and expertise in software engineering, AI/ML/DS, and Agile methodologies. Additionally, their respective roles and domains were taken into consideration. To ensure accuracy and precision, all participants consented to audio recordings of their interviews and meticulously documented key takeaways. Presented in Table 3 is a comprehensive summary of their information, encompassing their unique identifiers, roles, domains, project descriptions, and relevant experiences.

ID	Role	Domain	Area of focus	Experience		
				SE	AI/ML /DS	Agile
P01	Lead for AI for Strategy	Automotive	Computer vision and time series data and use transfer learning to adapt existing models.	4 Yrs	8 Yrs	6 Yrs
P02	ML Engineer	Energy	Deep learning algorithms and Image Processing.	6 Yrs	3 Yrs	2 Yrs
P03	Head of Applied Data Science	Chemical Industry	Machine learning, data science, IoT and optimization problems.	4 Yrs	8 Yrs	6 Yrs
P04	Data Science Engineer	Business Consulting and Services - Transportation	Machine learning, forecasting, statistical modelling, and more.	2 Yrs	2 Yrs	4 Yrs
P05	AI/ML Project Manager	Automotive	Data analytics and machine learning.		4 Yrs	½ Yr
P06	Global Head of Data Science	Human Resources Services	Data analytics and machine learning.	10+ Yrs	20+ Yrs	20+ Yrs

3.2.3 Literature Review

Additionally, an intensive literature review was conducted to comprehensively understand the existing knowledge and theoretical foundations related to the research topic. This review included academic journals, conference proceedings, books, and reputable online sources to ensure a robust and up-to-date subject matter analysis. Chapter 2 contains the literature review, while Chapters 4 and 5 summarise the challenges and recommendations identified in the literature review.

3.2.4 Data Analysis

The study used a qualitative approach to analyse data, which involved finding meaningful insights and patterns from interview data. The research was performed by carefully analysing collected interview transcripts and aimed to identify central themes, practices, and discoveries that could aid in developing a tailored agile methodology for AI/ML and Data Science projects. No coding or statistical analysis was used in this process. Instead, the focus was on capturing the participants' experiences, perspectives, and insights.

The analysis began with thoroughly reading the interview transcripts to comprehensively understand the participants' narratives. Then, the researcher manually summarised the significant statements, ideas, and recurring themes. The summarised data were systematically reviewed and grouped into broader themes and sub-themes from the interviews. These themes provided a comprehensive overview of the participants' perspectives, challenges, requirements, and suggestions regarding developing a customised agile framework for AI/ML and Data Science projects.

A qualitative data analysis approach allowed for the exploration of the participants' rich insights and experiences. It provided valuable qualitative evidence to inform the customised agile framework's development and validate its relevance and applicability in the context of AI/ML and Data Science projects.

Overall, the data analysis phase focused on extracting and synthesising the participants' perspectives and experiences related to the research topic. The findings from this qualitative analysis will serve as a foundation for refining and validating the customised agile framework in subsequent stages of the research.

3.2.5 Evaluation

Implementing a customised Agile methodology in a real-world project requires significant resources, time, and collaboration from various stakeholders. Due to possible disruptions, resource allocation, and organisational complexities, this is not feasible within the scope of this research study. Instead, a hypothetical project scenario was presented to evaluate the methodology. This will enable systematic evaluation and valuable feedback collection from participants. Although there are limitations to using a hypothetical scenario, it allows for a rigorous evaluation process crucial for refining and validating the soundness of the methodology. Feedback and insights gained from participants were used to improve the suggested approach, making it more adaptable to real-world projects in the future. This approach is a starting point for creating a comprehensive methodology that can be further tested and improved in practical settings.

4 Problem Analysis and Requirements

AI/ML and Data Science practitioners and project managers were interviewed to understand their challenges when using conventional Agile methodologies. Also, existing literature was reviewed on the topic. This chapter summarises the insights gained from both the interviews and literature review.

4.1 Insights from Interviews

The analysis started with investigating the challenges faced by AI/ML and Data Science projects using classic Agile methodologies. The interviews with AI/ML, DS practitioners, and project managers helped us gain valuable insights into their experiences and struggles. Through that, common themes and concerns were identified, such as difficulties in task estimation due to the unique characteristics of AI/ML model development, stakeholder expectations, and data-centric tasks.

To provide further context, Table 3 summarises the interviewees' project backgrounds.

	P1	P2	P3	P4	P5	P6
Methodology	Lean Data Science/ Kanban	Scrum	Lean Data Science/ Kanban	Kanban	Scrum/ Kanban	Scrum/SAFe
Roles	• Product Owner • Team Lead • Team (ML/Data engineers, Data scientists, SE, UX)	• Product Owner • Scrum Master • Team Lead • Team (ML/Data engineers, Data scientists)	• Product Owner • Team Lead • Team (ML/Data engineers, Data scientists, SE, UX)	• Data Scientist/ML Engineer • Data Engineer	• Product Owner • Project Manager • Team Lead • -Team (ML/Data engineers, Data scientists, SE, UX)	• Product Owner /Scrum Master • Senior ML Engineer • Team (ML/Data engineers, Data scientists, SE, UX)
Product Backlog Items (PBI)	Hypothesis	Use Cases, Tasks	Hypothesis	Tasks	Tasks	Epic, User Story, Tasks
Agile Practices	• Daily / Weekly Stand-up	• Daily Scrum • Scrum of Scrum • Backlog Planning • Sprint Planning • Retros - pective • Sprint Show & Tell	• Daily Stand-up • Sprint Review	• Weekly Update Meeting	• Weekly Update Meeting • Progress Review	• Daily Stand-up • Sprint Planning • Retros - pective • Sprint Review

Table 3 - Project Background of Interviewees

None of the participants are using a single standard project delivery methodology. They all have added customisation on top of their adopted primary methods. P1, P3, and P4 used Scrum earlier and adopted Kanban due to its flexibility. Table 4 shows the summary of the challenges mentioned by the interviewees.

	Challenge 1	Challenge 2	Challenge 3	Challenge 4
P1	Due to rapid development cycles and smaller projects, Scrum is too complicated/less flexible and time-consuming to set up.	It is difficult to estimate time and involves high unpredictability in the project's first phase with the problem statement and data exploration.	Consider alternative solutions before using ML for new projects or data analysis. This requires more stakeholder involvement and longer exploration tasks.	It is hard to manage expectations with stakeholders due to the uncertain and unpredictable nature of the AI/ML & DS projects.
P2	Data collection and cleaning in ML/DS projects take up a significant amount of the project timeline, making it difficult to show progress to stakeholders in each sprint, which delays visible outputs.	The absence of dedicated testers, requiring ML engineers to test their work, makes it challenging to estimate completion times accurately.	Stakeholders may not fully understand the intricacies of the technology, making it challenging to set clear timelines and expectations.	This approach is to use Agile methodology without strictly adhering to predetermined timelines. This helps to prioritize flexibility and continuous improvement over time-boxed deliveries.
P3	It is difficult to predict the timelines and determine the output in the earlier stages of the project, which include understanding the data, working with different datasets, and building and experimenting with models.	Due to the research nature of AI, ML, and DS projects, it is hard to share an increment of product with stakeholders within a predetermined timeline.	There is a need to integrate business stakeholders and their roles in the agile process from the beginning of the project and ensure effective communication and understanding of the nature of ML projects.	
P4	Due to the uncertain and evolving nature of ML projects, where the workflow may change during data exploration and model experimentation, it is challenging to apply Agile methodologies such as Scrum effectively, as they are less flexible to accommodate such changes midway through the project.	With the nature of ML projects, it is challenging to have deliverables at the end of defined periods, as progress may be more continuous, and the final deliverable may come after an extended development period.	Working with Scrum eventually led to spending more time on administrative tasks and defining tasks than actual development. The client's changing requirements and difficulty committing to sprints prompted a shift from Scrum to Kanban to maintain visibility and flexibility in task management without predefined sprints.	

P5	People often have unrealistic expectations of machine learning models, but their success depends on the quality and quantity of the data, which requires ongoing attention and skill to manage.	Technical experts in ML projects need substantial input from business stakeholders to understand their challenges and requirements. Encouraging effective communicatio n and collaboration between ML experts and business is crucial to deliver better solutions.	During project kick-off, much effort is invested with stakeholders to set realistic expectations about risks and the likelihood of achieving their desired outcomes and make them aware of both the benefits and limitations of the project due to data dependencies.	
P6	The organization faces challenges in connecting agile practices with classical portfolio management and project initiation, leading to difficulties in managing the entire agile initiatives at the portfolio level.	Data availability is crucial for starting machine learning modelling, and delays in obtaining data from external parties can present significant challenges as it hinders progress.	Data quality is a significant challenge in ML projects. The model's effectiveness heavily relies on the quality of the raw data, and often, data needs extensive preparation before it can be used effectively in the model processes.	

Table 4 - Challenge Summary from Interviews

4.2 Identification of Challenges

The interviews uncovered various obstacles encountered by Agile AI & ML projects. A summary of these challenges and the corresponding participant who brought it up can be found in Table 5 below. These hurdles comprise a range of concerns, such as complicated data management, business involvement, managing expectations, and adaptability of agile methodologies. All the interview participants were in favour of utilising the agile approach. However, they had differing opinions on various aspects, such as methodology, practices, and communication. Based on the interviews, seven common challenges were identified, which project teams encounter when utilising Agile methodologies for AI/ML and data science projects.

Challenge	**P1**	**P2**	**P3**	**P4**	**P5**	**P6**	
1. Challenges emerged in the data collection and data preparation phase	✓	✓	✓	✓	✓	✓	**100%**
2. Difficulty in forecasting time for exploration tasks in data exploration and model training phases	✓	✓	✓	✓		✓	**83%**

3. Limited flexibility for ad hoc tasks with a Sprint setting	✓	✓		✓	✓		**67%**
4. Unrealistic expectations from the key stakeholders	✓	✓	✓		✓		**67%**
5. Lack of business stakeholder involvement in project execution	✓		✓		✓		**50%**
6. Unable to show a shippable product at the end of the sprint		✓	✓	✓			**50%**
7. Connecting agile practices with classical portfolio management and project initiation						✓	**17%**

Table 5 - Challenge Analysis

4.3 Insights from Literature

Through interviews with professionals, seven challenges were identified that needed to be addressed. A comprehensive literature review of academic and industry research on Agile approaches in AI/ML and Data Science projects was conducted to understand these challenges better. This review included academic publications, conference materials, industry reports, and relevant books. Examining these sources confirmed the gaps and obstacles identified during the interview process. This literature analysis provides a comprehensive understanding of the issues at hand. The review of literature offers an opportunity to delve deeper into each challenge and learn from the insights of numerous authors. The summary of the literature review can be found in Table 6.

Challenge/Theme	Research Title	Summary/Excerpts/Finding	Author(s)/Reference
Challenges in the data collection and data preparation phase	A review and future direction of agile, business intelligence, analytics and data science.	In the field of Data Science and analytics, data is the main resource used. Data is constantly changing and organic, particularly with data sources' unstructured and undefined nature. Adapting to change remains crucial in both BI and DS development processes.	(Larson & Chang, 2016)
Challenges in the data collection and data preparation phase	A Taxonomy of Software Engineering Challenges for Machine Learning Systems: An Empirical Investigation	Developing an AI platform comes with several challenges, including handling data drifts in uploaded data, invalidation of models due to changes in data sources, and the need to monitor models in production for staleness.	(Lwakatare et al., 2019)

Challenges in the data collection and data preparation phase	An Agile Software Development Life Cycle Model for Machine Learning Application Development	Machine learning model development requires high-quality, relevant data. Engineers need to collect, clean, augment, and label the data. Understanding the data is crucial for successful data engineering and model debugging. Data consistency is also important for machine learning but can be challenging for systems that require manual labelling. This takes a longer time than the regular agile cycle.	(Ranawana & Karunananda, 2021)
Challenges in the data collection and data preparation phase	Agile4MLS—Leveraging agile practices for developing machine Learning-Enabled systems: An industrial experience.	ML development is experimental because there's no universally accepted best algorithm. An algorithm's effectiveness depends on accuracy, performance, and data availability. This is why ML development is iterative to ensure optimal results.	(Vaidhyanathan et al., 2022)
Challenges in the data collection and data preparation phase	How does Machine Learning Change Software Development Practices?	ML system requirements are often uncertain compared to non-ML systems. This is because they focus on improving decision-making processes and goals rather than detailed functional descriptions. Since the requirements of ML systems are data-driven, various inputs may produce different outcomes.	(Wan et al., 2021)
Challenges in the data collection and data preparation phase	How Do Engineers Perceive Difficulties in Engineering of Machine-Learning Systems? - Questionnaire Survey	There is a lot of uncertainty surrounding how the system will react to untested input data, especially when there is a significant change in behaviour due to a small input change (known as adversarial examples). The system's behaviour is greatly impacted by the training data it receives.	(Ishikawa & Yoshioka, 2019)

Challenges in the data collection and data preparation phase	Managing artificial intelligence projects: Key insights from an AI consulting firm	Agile logic is a process that revolves around feature or scope changes, which customers typically initiate. On the other hand, when making changes to tasks, data science considerations and intermediate data output are primarily what drive AI workflow logic.	(Vial et al., 2022)
Limited flexibility for ad hoc tasks with a Sprint	Agile big data analytics: AnalyticsOps for data science	When implementing an agile methodology for DS, providing clear guidance on essential activities and quickly moving towards an MVP product while keeping other enhancements for later iterations was a challenge. The analytics lifecycle was traditionally viewed as a waterfall process, with each step proceeding in sequence. Adopting an agile approach meant completing some activities from each step within a given sprint, a significant change from the traditional approach where changes upstream would require rewinding to that step and proceeding through the waterfall process again.	(Grady et al., 2017)
Limited flexibility for ad hoc tasks with a Sprint	Challenges of Integrating Data Science Roles in Agile Autonomous Teams	Experimenting and exploring the data was highlighted as important for data scientists to thrive. Participants explained that an important part of their job is to test and explore different hypotheses. If their work environment is too rigid, it becomes difficult for them to do their job.	(Hukkelberg & Berntzen, 2019)
Limited flexibility for ad hoc tasks with a Sprint	Managing artificial intelligence projects: Key insights from an AI consulting firm	When following Agile methodology, the practice is to keep iteration lengths consistent and prioritise completing specific tasks rather than partially finishing them. However, AI experimentation can lead to changes during iterations, making it difficult to formalise the content of each iteration,	(Vial et al., 2022)

		finish tasks within the iteration timeframe, and maintain consistent sprint durations.	
Difficulty in forecasting time for exploration tasks in data exploration and model training phase	An Agile Software Development Life Cycle Model for Machine Learning Application Development	Developing machine learning models involves trial and error to find the ideal combination of features and parameters. Engineers refine and fine-tune the model through multiple rounds of development and data and feature engineering.	(Ranawana & Karunananda, 2021)
Difficulty in forecasting time for exploration tasks in data exploration and model training phase	Analysis of Software Engineering for Agile Machine Learning Projects.	ML team A had more backlog issues than non-ML team B, indicating that estimating task duration for ML projects is harder. Unlike conventional software projects, ML projects involve applying various algorithms to data for predictions or recommendations, and accuracy can vary. Data type also plays a crucial role, as one algorithm may not work well for a different data type.	(Singla et al., 2018)
Difficulty in forecasting time for exploration tasks in data exploration and model training phase	Analysis of Software Engineering for Agile Machine Learning Projects.	It is difficult for many machine learning projects to predict the accuracy of a classification task, a regression, or the relevance of a recommendation since it ultimately depends on the data being used.	(Singla et al., 2018)
Difficulty in forecasting time for exploration tasks in data exploration and model training phase	Data Science Methodologies: Current Challenges and Future Approaches.	In data science projects, there are often many uncertain factors that can make the process challenging. Teams may need to go back and forth to find the right analysis tools, programs, and parameters, and a lot of trial and error may be involved. This can make it difficult to set realistic expectations and timelines for completing the project. Some experts have also pointed out that it can take time to fully understand the scope of the project and	(Martinez et al., 2021)

		the business goals at the outset.	
Difficulty in forecasting time for exploration tasks in data exploration and model training phase	How does Machine Learning Change Software Development Practices?	Planning for tasks in ML development can be challenging to ensure accuracy.	(Wan et al., 2021)
Difficulty in forecasting time for exploration tasks in data exploration and model training phase	How Do Engineers Perceive Difficulties in Engineering of Machine-Learning Systems? - Questionnaire Survey	The early stage of ML-based systems challenges customer decision-making due to the inability to estimate accuracy beforehand. Unrealistic expectations can lead to unnecessary costs and disappointment.	(Ishikawa & Yoshioka, 2019)
Difficulty in forecasting time for exploration tasks in data exploration and model training phase	Managing artificial intelligence projects: Key insights from an AI consulting firm	The AI workflow involves sequential dependencies, feedback loops, and multiple data exploration cycles. Fine-tuning the mathematical models is heavily influenced by the nature and quality of data. This makes predicting, planning, and managing the experimentation cycles challenging.	(Vial et al., 2022)
Unable to show a shippable product at the end of the sprint	An Agile Software Development Life Cycle Model for Machine Learning	According to reports, developing a functional production-grade prototype for most ML projects typically takes at least six months.	(Ranawana & Karunananda, 2021)

	Application Development		
Unable to show a shippable product at the end of the sprint	An Improved Agile Framework For Implementing Data Science Initiatives in the Government.	In certain ML/DS projects, observing noticeable results and patterns in the data may take several months. This is a challenge in these projects because data analytics is a creative process that may need multiple iterations to produce the desired outcomes.	(Qadadeh & Abdallah, 2020)
Unable to show a shippable product at the end of the sprint	Managing artificial intelligence projects: Key insights from an AI consulting firm	Agile principals frequently emphasise the importance of delivering a working product as the main indicator of progress. On the other hand, AI workflow logic is structured around conducting mini-experiments and forming hypotheses. However, the intermediate outputs of the AI workflow process do not allow for the regular delivery of tangible solutions	(Vial et al., 2022)
Business Stakeholder Involvement	A review and future direction of agile, business intelligence, analytics and data science.	To achieve faster in analytics and machine learning, it is important to prioritise discovery and iteration from the start. This requires increased interaction and collaboration to gain valuable insights.	(Larson & Chang, 2016)
Business Stakeholder Involvement	A review and future direction of agile, business intelligence, analytics and data science.	In order to discover information efficiently, data scientists and business stakeholders collaborate to select the appropriate data sources. Throughout the data processing and interim results stage, frequent communication with stakeholders is crucial to ensure the accuracy and direction of the project. Collaboration remains a top priority in data science, particularly when conducting descriptive analysis.	(Larson & Chang, 2016)

Business Stakeholder Involvement	A Taxonomy of Software Engineering Challenges for Machine Learning Systems: An Empirical Investigation	One of the challenges in formulating the ML problem is the need to establish a benchmark or baseline against which the accuracy and performance of the ML model can be evaluated and optimised. While various data tools can be used to aggregate and structure the data, important design decisions and trade-offs in model creation often require input from domain experts, such as identifying useful features. Although the models are not yet deployed, they provide valuable feedback to the experts about the potential impact of suggested features.	(Lwakatare et al., 2019)
Business Stakeholder Involvement	A Taxonomy of Software Engineering Challenges for Machine Learning Systems: An Empirical Investigation	When developing ML components, implementing the end-to-end ML pipeline can present challenges, particularly when developing an effective experimentation infrastructure. This infrastructure is necessary for evaluating ML models' performance improvements and impact, using metrics that focus on business outcomes rather than algorithms. However, designing and conducting multiple experiments on an ongoing basis is a complex task.	(Lwakatare et al., 2019)
Business Stakeholder Involvement	Agile big data analytics: AnalyticsOps for data science	Another challenge was predicting step durations for analytic systems operating in the Complex domain. Data sets can differ from documentation due to undocumented changes and may need to be cleaned. Data cleansing is an open-ended task unless well-managed and governed within the organisation.	(Grady et al., 2017)

Business Stakeholder Involvement	An Improved Agile Framework For Implementing Data Science Initiatives in the Government.	To align business strategies with analytics problems, it is crucial to convert business objectives into data mining. Understanding available data sets, resources, and variables is critical for defining problems and asking questions that require business involvement.	(Qadadeh & Abdallah, 2020)
Business Stakeholder Involvement	Beyond the Hype: Why Do Data-Driven Projects Fail?	From the responses received, it is clear that understanding business goals and user needs is crucial for successful data-driven projects. Most participants and six out of ten experts agree that a lack of business and user understanding is critical to project failure.	(Ermakova et al., 2021)
Business Stakeholder Involvement	Data Science Methodologies: Current Challenges and Future Approaches.	Poor coordination, collaboration, and transparent communication are issues between the client, analytics team, and IT department. Lack of support from the business side and domain expertise information hinders good results. Data analytics teams and scientists need help to work efficiently with IT and business agents.	(Martinez et al., 2021)
Business Stakeholder Involvement	Data Science Methodologies: Current Challenges and Future Approaches.	The lack of involvement by the business side can also be caused by a lack of understanding between both parties: data scientists may not understand the domain of the data, and the business is usually not familiar with data analysis techniques.	(Martinez et al., 2021)
Business Stakeholder Involvement	Exploring the resources, competencies, and capabilities needed for successful machine learning projects in digital marketing	The Marketing management team helped with ML development by understanding competitive needs and integrating with Digital Marketing (DM) systems for data collection. The DM team provided data availability insight and efficiently managed data to train the algorithm, making their role crucial for project feasibility.	(Blomster & Koivumäki, 2021)

Business Stakeholder Involvement	How does Machine Learning Change Software Development Practices?	ML practitioners often have less frequent communication with their clients.	(Wan et al., 2021)
Expectation Management with Stakeholders	A review and future direction of agile, business intelligence, analytics and data science.	In order to effectively utilise Business Intelligence(BI) and Data Science, a discovery process is necessary to determine customer expectations. Without clear expectations established beforehand, incorporating contracts into the development of BI/DS would become easier.	(Larson & Chang, 2016)
Expectation Management with Stakeholders	An Improved Agile Framework For Implementing Data Science Initiatives in the Government.	Missing business involvement and expectation management in project initiation can lead to confusion and project failure. Manage client expectations and ask important questions to avoid overlooking essential insights in the data. Consider the business, stakeholders, individual requests, available solutions, value, and project measurement.	(Qadadeh & Abdallah, 2020)
Expectation Management with Stakeholders	An Improved Agile Framework For Implementing Data Science Initiatives in the Government.	It is important to note that not all data science projects result in success. Therefore, the sponsors of these projects should be mindful of the potential risks associated with each phase. For instance, the dataset may not contain the necessary attributes to construct a useful model.	(Qadadeh & Abdallah, 2020)
Expectation Management with Stakeholders	Data Science Methodologies: Current Challenges and Future Approaches.	Many businesses expect data analytics teams to work miracles with little input. The high expectations set up by ML and deep learning techniques have induced a misleading perception that these new technologies can achieve whatever the business suggests at a very low cost, which is very far from reality.	(Martinez et al., 2021)

Expectation Management with Stakeholders	Challenges of Integrating Data Science Roles in Agile Autonomous Teams	Educating the role of a data scientist and their contributions is crucial. Setting and managing expectations, both internally and externally, is essential. The team lead or manager may have a different perspective than a data scientist and varying opinions on what is valuable. For instance, if an ML project does not succeed after two weeks, the data scientist may still find value in identifying unsuccessful algorithms and learning from the experience.	(Hukkelberg & Berntzen, 2019)
Expectation Management with Stakeholders	Failure of AI projects: understanding the critical factors.	Unrealistic expectations in AI projects can hinder adoption. Some labelled as AI don't use AI tech, leading to misunderstandings. Overly ambitious projects with broad scopes lack focus, making achieving goals difficult.	(Westenberger et al., 2022)
Expectation Management with Stakeholders	How Do Engineers Perceive Difficulties in Engineering of Machine-Learning Systems? - Questionnaire Survey	In this study, many comments show a lack of customer understanding, with nearly half mentioning high expectations for accuracy, ease of use, and functionality with limited data.	(Ishikawa & Yoshioka, 2019)

Table 6 - Literature Summary of Identified Challenges

4.4 Key Challenges Analysis

After conducting the interviews and the literature review, six critical challenges were identified. These challenges emerged as the participants' most frequently mentioned and significant issues.

The 5 Whys technique was used to uncover the root causes of these challenges through careful analysis of the data collected from interviews and literature review. The investigation revealed that the complexity and uncertainty associated with these challenges could be traced back to a common source: data dependency. Stakeholder engagement and transparency challenges often stem from an inadequate understanding of the project's nature and technical details. Additionally, the level of involvement and roles business stakeholders play can vary throughout the agile process, further complicating matters.

Based on this crucial insight, the challenges were categorised into two key challenges. A better understanding was gained by classifying the challenges, laying the foundation for developing a custom Agile methodology for AI/ML and DS projects. The summarised challenges are depicted in Figure 3, which visually represents the core obstacles aimed to address.

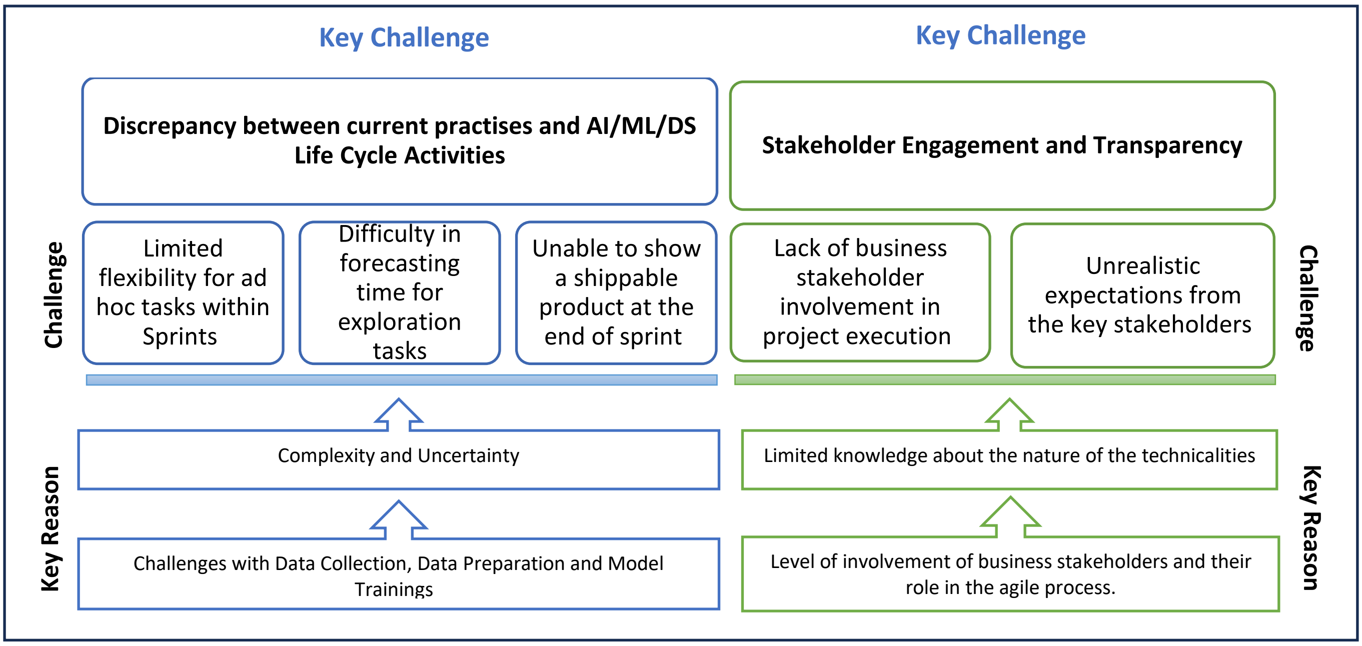

Figure 3 - Key Challenges - Cause Summary

4.4.1 Discrepancy between current practises and AI/ML/DS Life Cycle Activities.

a. Limited flexibility for ad hoc tasks with a time-box approach: Agile methodologies such as Scrum utilise time-boxed iterations that may not be compatible with the exploratory and uncertain aspects of AI/ML/DS projects. The fixed time constraints can impede the team's ability to adapt to ad-hoc challenges and explore options effectively. Several scholars, including Uysal (2022), J. Saltz and Sutherland (2019), and Singla et al. (2018) have also emphasised this challenge in their research. Estimating what can be accomplished within a sprint while using Scrum methodology for ML and DS projects can be difficult (Uysal, 2022). Fixed sprint durations can result in unrealistic and irrelevant backlog items that do not align with the project's requirements (Uysal, 2022). This implies that tasks such as exploratory data analysis and model evaluation may necessitate larger or smaller backlog item segments (Uysal, 2022). Data scientists and ML engineers face significant challenges when working on a sprint schedule, according to a study by Hukkelberg and Berntzen (2019). Due to the nature of their work, which is based on hypotheses, it may not always be possible to deliver immediate value from a management and team lead perspective.

P1, P2, P4, and P5 also brought up this challenge. P2 has highlighted that they are currently adapting scrum methodology and often face challenges with recording ad-hoc tasks emerging when doing exploratory work. Often, they do not record those in the tracking tool as it will affect the sprint scope. There are some challenges when using a sprint-based framework in Data Science. For instance, according to Saltz et al. (2017), estimating a task's length is difficult. This can create problems determining what can be accomplished within a sprint (J. et al., 2019).

Additionally, Scrum's fixed-length sprints can be problematic because they may force the team to include unrelated work items in an iteration (J. et al., 2019). This, in turn, could delay feedback from exploratory analysis and hinder the prioritization of new work (J. et al., 2019). According to a 2019 study by Brasjo and Lindovsky, most ML teams use sprints but do not strictly follow the Scrum process. Companies with prior software development experience tend to implement Scrum throughout their organisation for ML projects. Breaking Scrum rules is sometimes necessary for effective management of ML projects, such as sharing important insights internally immediately. However, strict adherence to sprint length and reluctance to share intermediary results, as advised by Scrum creators, can hinder efficient project execution in AI/ML projects.

Interview participant P4 also confirmed this. They had to adjust to a sprint-based methodology due to client preference. However, they spent considerable time on administrative tasks, which did not significantly impact the work as expected. Additionally, there was much time spent on planning tasks instead of performing them in real-time and identifying what was being done as it was being done. Sprints do not provide the flexibility to coherently complete smaller or longer logical chunks of work in the ML/DS context.

To tackle this challenge, P1, P3, and P4 have adopted rapid iteration cycles that are not time-restricted. P1 and P3 use hypothesis-driven iteration, testing one hypothesis in one iteration. In contrast, P4 follows a more flexible approach with weekly progress reporting. When dealing with ML or DS experimentation procedures, it is essential to adjust sprint lengths to address this issue (Uysal, 2021).

b. Difficulty forecasting time for exploration tasks in data exploration and model training phase: Much exploration and experimentation are involved when working on AI/ML and data science projects, making it difficult to predict how long these tasks will take accurately. Many machine learning projects cannot predict how well the accuracy of a classification task, regression, or relevance of a recommendation will be since it depends on data. Singla et al. (2018) found that ML projects have a higher average number of issues sent to the backlog, indicating that predicting task duration is more challenging due to more exploratory and research tasks (Singla et al., 2018). This also causes uncertainty in outcomes compared to conventional software projects (Singla et al., 2018). During their retrospective meetings, the ML project teams used a higher number of words such as "overestimation," "unexpected work," "delayed," "ad-hoc work," "waiting for," "not accounted for," "unplanned tasks," "more than expected," and "irregular" in the "what went wrong" section (Singla et al., 2018). This suggests there may be less certainty in the tasks due to the exploratory nature of machine learning-related tasks. (Singla et al., 2018). Managing and versioning data for machine learning applications is a complex and challenging task that differs from other forms of software engineering. (Amershi et al., 2019). Understanding machine learning involves understanding data (Amershi et al., 2019). Managing and working with data involves more effort and rigour than dealing with software code. Amershi et al. (2019) stated that it is a complex and distinct process. Developing machine learning models involves designing, training, evaluating, deploying, and monitoring them using data (Ranawana & Karunananda, 2021). Since machine learning systems are non-deterministic, sequential development methods can be challenging (Ranawana & Karunananda, 2021).

P1, P2, P3, P4, and P6 have also emphasised this challenge. As P6 mentioned, data availability is crucial for starting ML modelling, and delays in obtaining data from external parties can present significant delays in progress. This is beyond the team's control; they need data to start properly. When the data is available, data quality causes a significant challenge. The model's effectiveness heavily relies on the quality of the raw data, and often, the data needs extensive preparation before it can be used effectively in the model's processes. In the data preparation and model training phases, the team could come across missing values, missing parameters, or data not matching the problem.

c. Unable to show a shippable product at the end of sprint: AI/ML projects differ from conventional software development because they may not always produce a shippable product after each sprint. They rely heavily on data and involve an iterative model development process. Experimentation is crucial when developing ML models (Ranawana & Karunananda, 2021). Since these models rely on various parameters, it is necessary to try different combinations and features to find the best results (Ranawana & Karunananda, 2021). Engineers go through multiple rounds of development to identify data dependencies and determine the ideal set of features for optimal model performance (Ranawana & Karunananda, 2021). Refining and tuning the model requires further experimentation through data and feature engineering (Ranawana & Karunananda, 2021). Qadadeh and Abdallah (2020) conducted a study highlighting the challenge of achieving higher trust in data analytics projects. It was found that such projects take months to show results and patterns in the data (Qadadeh & Abdallah, 2020). The study also identified that data analytics is a creative process that may require more than one iteration to yield the desired results, and the knowledge and skills of resources play a crucial role in the success of such projects (Qadadeh & Abdallah, 2020).

P2, P3, and P4 also confirmed this. In an ML project, the P2 suggests that data collection and cleaning account for 60 to 70 per cent of the work. If they plan to complete the project in 10 sprints, they will likely spend approximately four or five sprints on data collection and cleaning. This means they cannot show any interface or results to the end user until the fifth or sixth sprint. Regardless of their implementation, the project follows a waterfall method because the data collection stage is invisible to the end user. This poses a significant challenge for development. They have agreed with stakeholders that the end-of-sprint delivery artefact need not be a shippable product but could instead take the form of a graph, table, or accuracy metrics. P3 highlighted that the standard methodology used in software development differs from machine learning. Developing an ML model involves understanding the data through various data sets and building and experimenting with the model. Only at the end of this process can it be wrapped around a software engineering solution. The standard agile approach works well towards the end of the process, but it needs to be better adapted to the beginning of the process.

It has been discovered that the main issue causing difficulties in AI/ML projects is Data Dependency. This results in complicated and uncertain situations. Data quality and availability are crucial to the success of AI/ML models, making data dependencies a significant factor in executing projects. According to Ranawana and Karunananda (2021), data collection and preparation are the most significant challenges in developing machine learning models. No matter their experience level, data availability, collection, cleaning, and management are the top challenges that respondents face, according to a study by Amershi et al. (2019). This finding was highlighted as the key takeaway from their research. It is important to collect, clean, augment, and label data appropriately to develop an ML model. Engineers may need time to understand the data, which is crucial for successful data engineering and model debugging. Consistency in data is also important for machine learning, but it can be challenging to achieve in systems that require a lot of manually labelled data. Typically, the best approach is to start with sufficient data to train the machine learning model effectively. Machine learning processes are closely linked to data science and data mining workflows, reflecting the data dependency of ML model development.

According to Lwakatare et al. (2019), three main methods are used to develop software: requirements-driven, outcome-driven, and AI-driven. Creating AI-enabled applications that use machine learning (ML) techniques, such as deep learning (DL), involves creating ML models based on data (Lwakatare et al., 2019). Typically, multiple experiments are conducted before selecting the final ML model (Lwakatare et al., 2019). During the creation of these models, learning algorithms are applied to a dataset to train and evaluate their accuracy and performance (Lwakatare et al., 2019). Ishikawa and

Yoshioka (2019) also noted a high level of uncertainty regarding how the system will respond to untested input data, such as when a slight change in the input results in a radical behaviour change (known as adversarial examples). All six participants confirmed that the main source of complexity and uncertainty was attributed to data dependency.

The 5 Why Analysis of the main challenge 1, which is the discrepancy between current Agile practises and AI/ML/DS Life Cycle Activities, is illustrated in Figure 4.

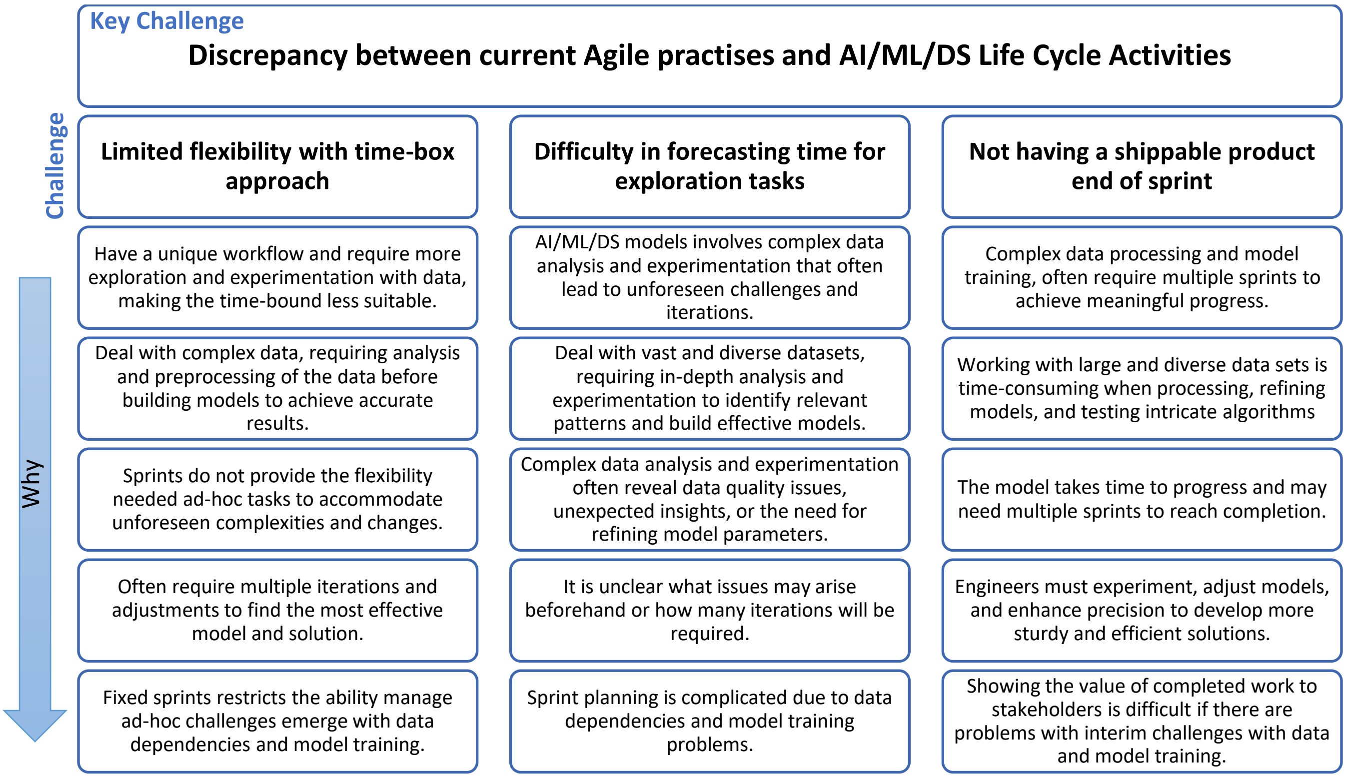

Figure 4 - Key Challenge 1- 5 Why Analysis

4.4.2 Stakeholder Engagement and Transparency

a. Business User Engagement: Executing AI/ML/DS projects requires efficient collaboration and engagement with business stakeholders. Nevertheless, establishing clear communication and understanding between technical teams and business users may be difficult due to the intricate nature of these projects and the expertise involved. According to a study conducted by Vial et al. (2022), the lack of customer knowledge and understanding of AI can hinder the success of AI projects, and the gap between those who know and those who do not know about AI is widening. This can result in a poor understanding of the data, and high opportunity costs for the customer and the company assisting them (Vial et al., 2022). Carter and Hurst (2019) emphasise the importance of business involvement in their book, "Agile Machine Learning." Chapter four focuses on aligning with the business and provides best practices for how maintaining business involvement can contribute to project success. In that chapter, one of the authors shares their experience working at Microsoft. They noted much separation between them and the businesspeople developing AI/ML products. At Microsoft, the "program manager" role involved interacting with businesspeople and translating their needs into requirements for the engineering team. The author indirectly received guidance from the Program Management (PM) team about what businesspeople wanted in the products. This level of indirection was convenient for both business and development. However, it also created another separation between the author, an engineering manager, and the businesspeople.

P1, P3 and P5 also brought up business stakeholder involvement. As P5 shared her experience, "Technical experts need input from businesspeople to understand their challenges and pain points. Workshops only provide limited information. The more experts know about business needs, the better they can deliver. Business departments are often busy and focused on getting working solutions, making it difficult to gather necessary information. This is an ongoing challenge that we aim to improve."

b. Expectation Management: Managing stakeholder expectations is especially important for AI/ML/DS projects since multiple factors can influence outcomes. It is essential to balance their expectations and be transparent about the project's progress and limitations to prevent miscommunication. In their study, Hukkelberg and Berntzen (2019) emphasise the importance of managing expectations within a team and throughout an organisation. They stress the need for dedicated time and effort to address different perspectives on value (Hukkelberg & Berntzen, 2019). The review highlights the potential disconnect between team leads/managers and data scientists/ML engineers regarding understanding value (Hukkelberg & Berntzen, 2019). While a data scientist may see value in a machine learning implementation that did not meet expectations due to the knowledge gained and lessons learned, a manager might view it as a failure and struggle to see any value. Therefore, the review suggests that integrating new roles into a team requires considering the various aspects and perspectives of value (Hukkelberg & Berntzen, 2019). Overall, the study emphasises the importance of managing expectations effectively and fostering a shared understanding of value to ensure successful collaboration within the team and throughout the organisation (Hukkelberg & Berntzen, 2019).

According to a study by Ishikawa and Yoshioka (2019), the primary concerns in early explorations revolve around customer decision-making. In traditional settings, this involves analysing requirements and specifications during the initial phase and conducting acceptance inspections in the final stage (Ishikawa & Yoshioka, 2019). However, with ML-based systems, it is only possible to estimate accuracy after, making this process difficult (Ishikawa & Yoshioka, 2019). Despite this, many customers have unrealistic expectations of AI/ML, leading to unnecessary costs, missed opportunities, and disappointment (Ishikawa & Yoshioka, 2019). Additionally, it was stated that if customers grasp the concept of imperfection, we can then address the primary challenges that arise from it (Ishikawa & Yoshioka, 2019). Many of these challenges stem from uncertainty, such as our inability to determine the value of something until it is built and the fact that functionality is imperfect and may not always work (Ishikawa & Yoshioka, 2019).

In their book, Carter and Hurst (2019) noted that many businesses that begin using machine learning-based solutions expect extraordinary results from their development teams. While machine learning can produce remarkable outcomes, businesses must recognise its limitations. One of the most significant challenges has been helping business professionals comprehend the functional nuances of a specific machine-learning model.

P1, P2, P3, and P5 also shared about this challenge. P1 and P5 have mentioned that business stakeholders may have high expectations of machine learning solutions, even viewing them as a "silver bullet." However, it is important to remember that other effective and simpler solutions may be available.

P2 shares that they have agreed upon the value beyond a shippable product, effectively sharing their progress after each sprint with the business. According to P5, they have solved their challenge by dedicating more time and effort to the project kick-off phase. They discuss everything upfront, including the risks associated with desired functionalities, such as Functionality A, who is responsible for those risks, and how to mitigate them. They communicate this information to clients to manage their expectations and ensure they understand the likelihood of success and potential risks. P5 also highlights the benefits and opportunities of the desired functionalities while acknowledging the limitations and potential downsides. By doing so, they provide a realistic picture of what clients can expect.

Figure 5 illustrates the 5 Why Analysis of the main challenge 2: Stakeholder Engagement and Transparency.

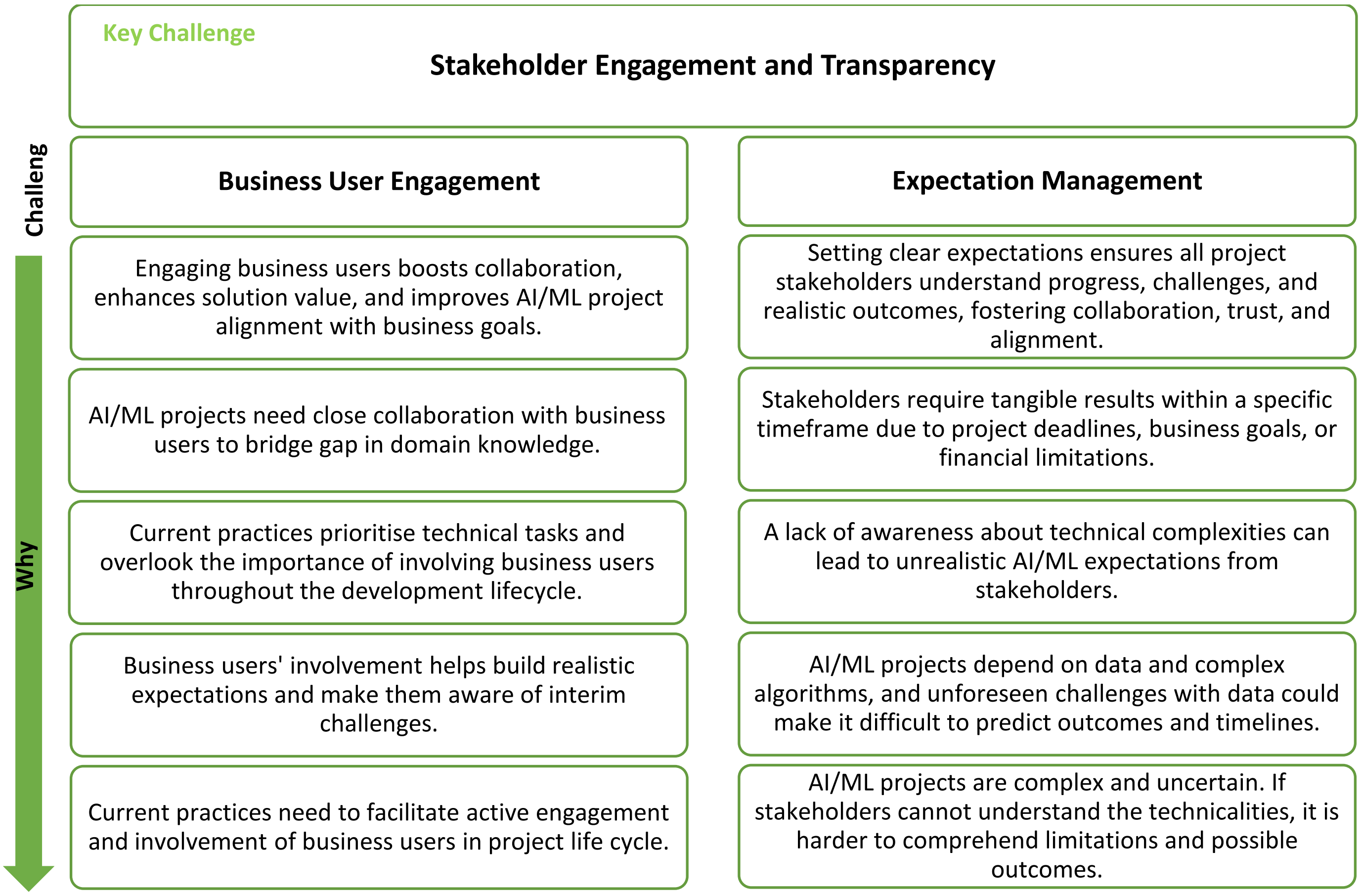

Figure 5 - Key Challenge 2 - 5 Why Analysis

5 Design and Development

This chapter outlines the steps to create the customised Agile approach incorporating principles from Kanban, Lean Startup, and Design Thinking. These methodologies were specifically chosen to tackle the unique challenges that arise in AI/ML and DS projects, such as the need for adaptability, iterative progress, and a customer-focused approach.

After conducting a thorough literature review, it was determined that adopting Kanban, Lean Startup, and Design Thinking would benefit the custom Agile approach in AI/ML and DS projects. These methodologies have been proven effective in managing complex and uncertain projects according to various academic and industry sources. In particular, Kanban's ability to promote continuous flow and adaptability has been highlighted in studies involving AI/ML and DS projects with fluctuating data requirements. Meanwhile, Lean Startup's focus on early validation and feedback has received praise in the AI/ML and DS community, where experimentation and rapid learning are essential. Finally, Design Thinking's user-centric approach has been well-received by researchers and practitioners, emphasising the importance of involving stakeholders throughout the development process. Table 7 shows an overview of the recommendations from the literature review.

Challenge/Theme	Reccomendation	Research Title	Author(s)/Reference
Limited flexibility for ad hoc tasks with a Sprint	Data science deals with unstructured data quickly acquired and stored for analysis. This eliminates the need for traditional design steps. The discovery process, which is typically done during design and development, is now moved to the beginning of the development cycle. Data analysis begins as soon as the data is acquired. The data is visualised interactively and iteratively to aid the discovery of insights. There is limited research on the application of Agile principles in this context. However, the available research suggests that Agile would align well with this approach. It is recommended that a "short-cycle Agile" approach be adopted to ensure faster results. Short-cycle Agile refers to faster and more flexible sprints.	A review and future direction of agile, business intelligence, analytics and data science.	(Larson & Chang, 2016)
Limited flexibility for ad hoc tasks with a Sprint	As a standard practice, it is important to log the current status of each subtask or story, not just the expected status. This is because certain tools may prevent updates to ad-hoc tasks during the middle of a sprint.	Analysis of Software Engineering for Agile Machine Learning Projects.	(Singla et al., 2018)

Unable to show a shippable product at the end of the sprint	Consider redefining what it means to be "done" for AI projects. Consultants often conduct AI experiments that yield valuable intermediate results, which help to validate design elements and move the project forward. Although these outcomes may not fit the traditional definition of project deliverables, they are still crucial for project teams to succeed.	Managing artificial intelligence projects: Key insights from an AI consulting firm	(Vial et al., 2022)
Expectation Management with Stakeholders	The goal is prioritising collaboration over creating a detailed plan for BI/DS projects. Detailed plans can be challenging to make due to limited high-level planning information. Collaboration can overcome this by defining expectations and improving communication among stakeholders.	A review and future direction of agile, business intelligence, analytics and data science.	(Larson & Chang, 2016)
Expectation Management with Stakeholders	Address communication and collaboration challenges in data science projects should include documenting and visualising the project's status.	Data Science Methodologies: Current Challenges and Future Approaches.	(Martinez et al., 2021)
Expectation Management with Stakeholders	To overcome challenges in decision-making with customers, trial-based processes can help refine assumptions and goals through experimentation.	How Do Engineers Perceive Difficulties in Engineering of Machine-Learning Systems? - Questionnaire Survey	(Ishikawa & Yoshioka, 2019)
Expectation Management with Stakeholders	Assess the customer's readiness and maturity level during initial evaluations. Conduct periodic checks using stage gates to re-evaluate the feasibility of the project and the customer. Sometimes, it may be best for both parties to halt or terminate the project.	Managing artificial intelligence projects: Key insights from an AI consulting firm	(Vial et al., 2022)
Business Stakeholder Involvement	To succeed in data science, it is crucial to understand the client's goals and prioritise them. Defining business objectives is essential, and design thinking principles can help.	An Improved Agile Framework For Implementing Data Science Initiatives in the Government.	(Qadadeh & Abdallah, 2020)

Business Stakeholder Involvement	Lack of business and user understanding is critical to project failure. To ensure success, it is essential to incorporate steps prioritising user and business understanding in the search for data-driven solutions.	Beyond the Hype: Why Do Data-Driven Projects Fail?	(Ermakova et al., 2021)
Business Stakeholder Involvement	Data analytics teams need input from business professionals to work effectively. The need for more understanding between the two can be bridged by an intermediary who understands both domains.	Data Science Methodologies: Current Challenges and Future Approaches.	(Martinez et al., 2021)
Business Stakeholder Involvement	Defining roles helps coordinate team members and stakeholders.	Data Science Methodologies: Current Challenges and Future Approaches.	(Martinez et al., 2021)
Business Stakeholder Involvement	Key factors for unsuccessful data-driven projects were business understanding and understanding user needs. Including additional steps for user and business needs is crucial. According to a data scientist from Airbnb, including Design Thinking principles in the CRISP-DM process can increase the success rate of data science projects by 5–10 times.	Beyond the Hype: Why Do Data-Driven Projects Fail?	(Ermakova et al., 2021)
Business Stakeholder Involvement	Recommending an approach for an AI project is, to begin with a Design Sprint, which involves a workshop led by Design Thinking principles.	Fusing Artificial Intelligence with Scrum Framework	(Ameta et al., 2023)
Business Stakeholder Involvement	Instead of searching for one person who can handle every aspect of a project (known as a unicorn), Consult takes a different approach. They match a business consultant with a data scientist to enhance communication and balance business and technical needs for their projects.	Managing artificial intelligence projects: Key insights from an AI consulting firm	(Vial et al., 2022)

Table 7- Literature Summary of Solution Recommendations

5.1 Solution Design

The Agile approach proposed for AI/ML and data science projects integrates principles from Kanban, Lean Startup, and Design Thinking. The aim of this study is not to create a completely new methodology. It was found that utilising agile methods is beneficial when working on AI/ML and DS projects. However, there is a need for a single methodology that encompasses the best practices. This study introduced improvements to Kanban for AI/ML and DS projects based on feedback and recommendations gathered from interviews and literature. Each methodology brings unique strengths that complement and enhance the Kanban way of working, resulting in a more adaptable, value-driven, and user-centric approach. Table 8 shows the summary of the challenge and solution mapping.

Challenge	Suggested Solution	How it Solves the Challenge	Integration with Kanban
Limited flexibility with time-box approach	Capability-based and experiment-driven iterations: Iterations are not time-boxed but capability-based and experiment-driven. Each iteration should not exceed 30 days.	Iterations are not time-boxed but experiment driven. This allows teams to create small experiments and rapidly iterate	By using a Kanban board, the team can visualise their workflow and have a clear understanding of the status of each item. Without any time-box constraints, the team can easily move tasks across different stages of the board as they are ready to be worked on or completed, providing a dynamic and flexible approach to managing their work.
Difficulty in forecasting time for exploration tasks	Iterations are defined by selecting the experiment that will be the team's focus	The experiment-based approach allows the team to prioritise items without estimating how many tasks to include in an iteration. This promotes adaptability and reduces the need for time forecasting during exploration tasks.	A Kanban board provides a clear and organised visualisation of the selected experiment and its tasks. It enables the team to swiftly identify and choose the most crucial tasks based on their value and significance. Forecasting becomes more adaptable and less rigid by iteratively exploring uncertainties and focusing on critical tasks.
Not having a shippable product by the end of a sprint	Experiment-driven iteration with Lean Startup cycle (Build, Evaluate, Learn): Each iteration is a single testable experiment	To achieve significant progress, focus on delivering a single experiment that can be tested in each iteration. This promotes a mindset of learning and adapting and allows for early feedback gathering.	Using a Kanban board with clear columns for various development stages is a great way to improve the visibility and transparency of the experiment's progress. This helps teams easily track the experiment, understand their position, and identify areas for improvement.

Business User Engagement	Lean Startup (LS) – Build, Evaluate, Learn Iteration: Emphasis on continuous communication and collaboration.	The LS approach emphasises the importance of regularly involving business users and stakeholders in the project. Creating frequent feedback loops makes it easier to address uncertainties more effectively. Focusing on finishing one experiment per iteration to make progress will promote a culture of learning and adaptation and allow for early feedback gathering.	The Kanban board is great for visualising ongoing work, increasing transparency and collaboration among the team and stakeholders, and can be used as a reference during progress meetings.
Expectation Management	Design Thinking for better alignment in the project initiation phase and Lean Star up for iterative problem-solving process	The team can cultivate empathy and thoroughly comprehend user requirements and concerns using Design Thinking. Involving stakeholders throughout the process and establishing practical expectations during project initiation and each iteration can boost alignment with user expectations and reduce unexpected issues. Additionally, Lean Startup helps sustain communication with business users and stakeholders, giving them a realistic overview of progress.	When the project is initiated, the Kanban board could be utilised to build a high-level backlog and prevent discrepancies between expected outcomes and actual results with better visualisation. This early detection can result in better goal setting, improved expectation management, and increased transparency.

Table 8 - Challenge - Suggested Solution Mapping

5.1.1 Kanban

Rationale: The Kanban method is a way to manage project workflows that promotes transparency and flexibility visually. The proposed Agile approach incorporates Kanban principles, focusing on visualising work items, managing flow, and limiting work in progress (WIP). After the first interview, it was found that 4 out of 6 participants (P1, P3, P4, P5) use Kanban or a Kanban-based approach for task management and that it works well. Additionally, 3 out of 6 participants (P1, P3, and P4) switched to Kanban after using Scrum, citing better flexibility as the reason for the change.

Kanban is an agile methodology that aligns with lean thinking and just-in-time scheduling. It has no specific process framework or designated roles, but meetings can be arranged as needed. Teams can use their process models if they adhere to Kanban principles. The core principles of Kanban include using a task board, visualising the workflow, limiting work-in-progress, and measuring and managing flow through quick feedback loops. Its goal is to maximise value while minimising waste, which is achieved by balancing work demands with the team's available capacity to avoid bottlenecks (Uysal, 2022).

Although the flexibility and freedom offered by Kanban can be beneficial, it can also present challenges in its implementation. The absence of a defined process structure can be advantageous as it permits teams to adapt Kanban to their existing organisational practices. However, it can also result in each team implementing Kanban differently, creating a lack of cohesion and consensus. This means that teams must establish their processes and artefacts when using Kanban. Hence, it is unsurprising that integrating Kanban with existing agile techniques can be complex, costly, and time-consuming (J. Saltz & Suthrland, 2019). Therefore, it is essential to support Kanban with additional practices since it does not explicitly specify a process framework.

According to a study by Vial et al. (2022), agile practices and methods are commonly used to develop AI applications across all project phases. During the initial phases, teams often employ a Kanban approach to allow for weekly iterations. This approach is particularly useful due to the exploratory nature of the work undertaken during these early phases. Furthermore, agile processes allow teams to identify and address issues early, such as underperforming models or lower-than-expected data quality. Participants in the study noted that the iterative nature of agile practices allows failure to happen more quickly, which can ultimately result in faster success (Vial et al., 2022).

Further, a study by J. S. Saltz et al. in 2017 tested various agile methodologies for data science projects. The experiments showed that Kanban was the most preferred methodology regarding ease of use, project results, team member satisfaction, and willingness to work on future projects. Further, Martinez et al. (2021) conducted a systematic review and compared various approaches to address communication and collaboration challenges in data science projects. As a result, they proposed recommendations that include documenting and visualising the project's status. This could be achieved through using a Kanban board for workflow management.

Conversely, Scrum is a well-established methodology with established working practices. However, it can be challenging to incorporate the necessary flexibility required for AI/ML and DS projects, which may result in changes to scrum principles. Therefore, it was decided to implement customisation using Kanban as the foundation.

Contribution: AI/ML and data science project tasks and requirements can often be unpredictable and ad hoc. However, using Kanban can allow teams to handle this uncertainty more effectively. By visualising tasks on Kanban boards, team members can easily see the status of each task and prioritise the most critical ones. Additionally, by limiting work in progress, the team can avoid overloading and improve efficiency.

5.1.2 Lean Startup

Rationale: The principles of Lean Start-up revolve around building, measuring, and learning. It promotes an approach that focuses on forming hypotheses and conducting experiments to validate assumptions while also being open to pivoting when necessary.

Eric Ries introduced the concept of Lean Start-up (LS) (Figure 6), which serves as a blueprint for managing a start-up (Ahmed et al., 2018). To succeed in fast-paced and uncertain environments, startups must iterate quickly based on customer feedback (Bortolini et al., 2018). The LS methodology, developed by Ries, provides a simple and efficient way for organizations to experiment and evolve their business model (Bortolini et al., 2018). Its core objective is to identify a suitable product-market fit by following the 'Build-Measure-Learn' feedback loop (figure 7) with a 'Minimum Viable Product' (MVP) (Ahmed et al., 2018). The first step in this process is to create a basic set of features that satisfy early users. Then, the team tests their hypotheses about these features and gathers feedback. This feedback is used to make evidence-based decisions on progressing the solution in subsequent iterations. This cycle continues until the product-market fit is achieved. The goal is to ensure that the solution developed by the team addresses the needs of the customer/user and proves successful through experimentation (Ahmed et al., 2018). The principle behind Lean Start-Up methodology can be applied to AI/ML project delivery to eliminate uncertainty by iterating often and failing early.

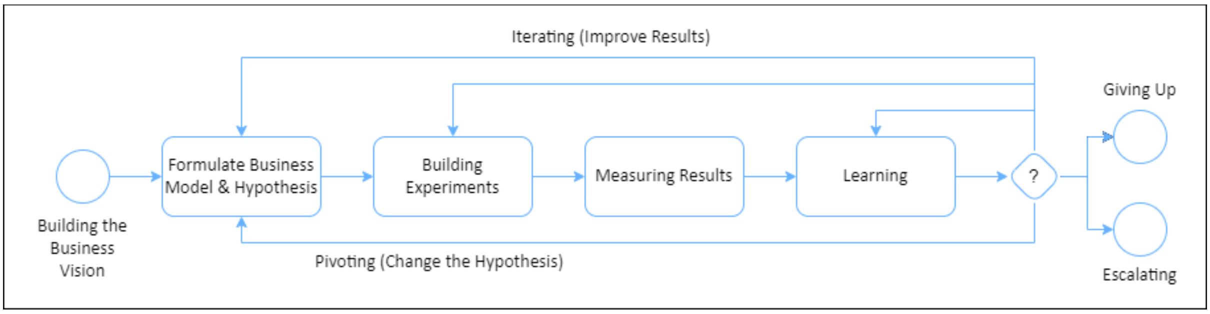

Figure 6 - Lean Strat-Up Model (Bortolini et al., 2018)

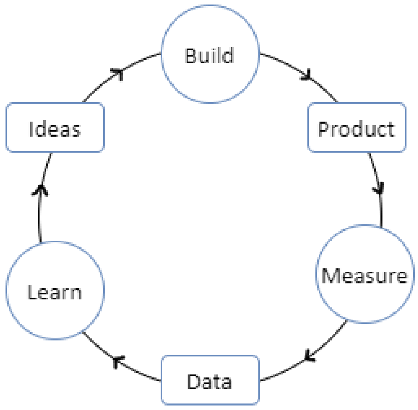

Figure 7 - Lean Start-up Build - Measure - Learn Cycle

Teams adhering to the Agile Manifesto's principles strive to involve customers throughout the development process, according to a study by Vial et al. from 2022. The ultimate objective is to gradually transfer ownership of the solution to the customers upon delivery of the MVP. In their study, a Senior AI consultant and agile product owner emphasised the importance of involving customers in experiments so that they feel involved and informed (Vial et al., 2022). This strategy promotes knowledge transfer and ensures that customers know the need for experimentation.

The proposed approach places a greater emphasis on the Lean Start-up Build-Measure-Learn cycle during the development stage. This provides a more comprehensive approach to developing a solution by collecting feedback from customers/users and learning from it. Valuable feedback can be obtained through testable experiments such as direct observation and customer/user discussions. Alternatively, machine learning algorithms can also provide accurate statistical data for feedback (Ahmed et al., 2018). By utilising these two forms of feedback, the development team can assess if they should proceed with their current direction, reevaluate the core concept of the solution, modify the dataset, or generate a new solution. The solution undergoes iterative development, with features added, removed, or modified based on feedback until a final solution is achieved. This is how the AI/ML lifecycle works, and its nature has been incorporated into the proposed approach.

Contribution: When delving into AI/ML and data science projects, there will inevitably be uncertainties and hypotheses that need constant validation. However, Lean Startup practices offer a solution through frequent interaction with business users and stakeholders, allowing for feedback gathering and project direction adjustment. This methodology and Agile practices create an effective feedback loop that promotes iterative improvements, ultimately reducing the risk of building irrelevant features or models.

5.1.3 Design Thinking

Rationale: *Design Thinking* is a user-focused strategy that emphasises understanding the needs and experiences of individuals. It prioritises empathy, creativity, and iterative problem-solving as key elements.

The initial task of a data scientist is to meet with the business user or client to comprehend their needs and requirements (Qadadeh & Abdallah, 2020). The primary objective of a data scientist is to identify the crucial objectives at the outset of the project that will impact its outcome. It is essential to avoid wasting time answering irrelevant questions, as this can lead to setbacks in project execution and is unacceptable in any organisation (Qadadeh & Abdallah, 2020). Without a clear understanding, no data mining technique can yield desirable results for business users (Qadadeh & Abdallah, 2020). Next to the data preparation, defining the business objective is the second most important time and effort aspect of an ML/DS Process (Qadadeh & Abdallah, 2020). Further in their study, Qadadeh and Abdallah (2020) suggested that approaches like *design thinking* should be followed to develop valuable and meaningful applications to achieve this goal.

Design thinking, shown in Figure 7, uses divergent and convergent techniques to understand a problem, create a solution, and successfully bring it to market (Koehnemann, 2023). Design Thinking is a problem-solving approach that involves creating useful solutions through ideation (Ahmed et al., 2018). The approach's fundamental principle is prioritising development teams' understanding of the end-users before creating a solution (Ahmed et al., 2018). They utilise techniques like empathising to gain insight into human behaviour and understand user needs (Ahmed et al., 2018). Developers first empathise with the user to identify their most pressing problems (Ahmed et al., 2018). Then, they prioritise these problems and brainstorm ideas to solve them (Ahmed et al., 2018). These ideas are tested and evaluated through experimentation before being implemented. Design thinking encourages us to consider new methods for evaluating the effectiveness of the initiatives. This involves considering factors such as customer appeal (Desirability), the feasibility of delivering the ideal solution (Feasibility), whether it is creating more value than cost and making a profit (Viability) and managing the solution for its expected market lifespan (Sustainability) (Koehnemann, 2023). The goal is to leverage the practices associated with the ‘Empathise’, ‘Define’, and ‘Ideate’ phases of Design Thinking during the early stages of AI/ML and Data Science projects. This approach can help make the most of the kick-off and workshops by applying design thinking principles. This approach enables technical experts and business users to share their thoughts openly and collaborate more effectively.

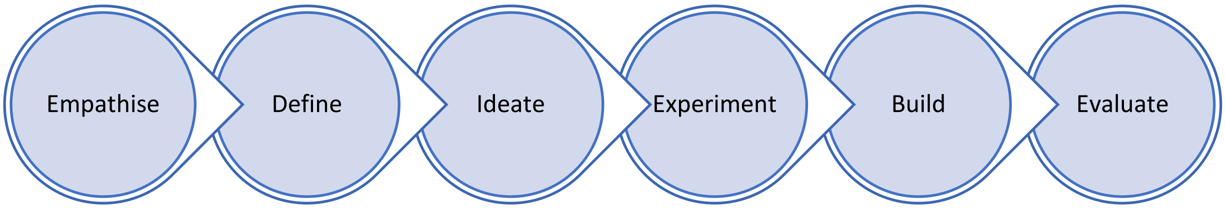

Figure 8 - Stages of Design Thinking (What Is Design Thinking & Why Is It Important? | HBS Online, 2022)

Amit Mishra from Oracle suggested using the Design Thinking phase to analyse and translate the business requirements into technical requirements (Qadadeh & Abdallah, 2020). These requirements should be quantitatively represented and connected to a traceability matrix (Qadadeh & Abdallah, 2020). The data mining process should not be linear, as the agility nature of it could potentially compromise the project's integrity (Qadadeh & Abdallah, 2020). Mishra claims that combining Design Thinking with System Engineering is an effective approach to generating innovative ideas and delivering reliable data science solutions with high-agility processes (Qadadeh & Abdallah, 2020).

A recent study by Mishra (2021) successfully developed an AI application for mineral processing using the Design Thinking methodology in their problem identification phase. This approach emphasises the importance of immersing oneself in the problem space, particularly in industrial AI innovation projects (Mishra, 2021). He further elaborates that the "no free lunch" theorem suggests that there is no one universal algorithm that can solve all problems (Mishra, 2021). Algorithms can perform well in certain tasks but not in others. Therefore, defining the problem accurately is crucial in finding an algorithm that performs exceptionally well (Mishra, 2021).

Contribution: To tackle challenging problems in AI/ML and data science, it is crucial to comprehend users' needs and devise innovative solutions. Design Thinking principles provide an effective framework for gaining a comprehensive understanding of end-users and stakeholders, thereby enabling the development of tailored AI/ML models and data-driven insights to address real-world challenges effectively. By incorporating Design Thinking practices, the Agile approach becomes more user-centric, resulting in solutions better aligned with user expectations and requirements.

5.1.4 Key Attributes

1. **Experiment-driven and Iteration-based Development**

   Adopting iteration-based development allows the project team to break down complex tasks into smaller, manageable experiments that can be incrementally developed, tested, and delivered. Agile methodology is closely related to AI/ML and DS due to their shared focus on small and rapid releases (Larson & Chang, 2016). This approach to analytics and data science projects is best delivered in increments, allowing businesses to evaluate the impact of changes and manage risk quickly (Larson & Chang, 2016). By providing more control and tangible results, an incremental approach enables customers to see the benefits of these changes (Larson & Chang, 2016). It is worth noting that the shorter the scope and cycle of data science, the quicker the results (Larson & Chang, 2016). The study by Larson and Chang (2016) highlights that short-cycle Agile, which refers to faster and more flexible sprint approaches, is successful due to its flexibility and swiftness. "Iterations" are cycles aimed at improving and revising the deliverable, while "increments" are scheduled as part of a roadmap or release plan tied to an overall business strategy (Larson & Chang, 2016).

According to Jurney's (2017) book, a data science team should prioritise overseeing several experiments simultaneously rather than only assigning tasks. Iteration and experimentation are crucial for gaining insights (Jurney, 2017). Through exploratory data analysis, valuable assets such as tables, charts, reports, and predictions can be produced (Jurney, 2017). Therefore, it is recommended to approach projects as experiments instead of tasks (Jurney, 2017). Considering the limited flexibility with fixed-length sprints, an iteration driven by an experiment was suggested. The iteration length could vary depending on the scope of the experiment. However, a single iteration could be between 1 week to 4 weeks. According to the Agile Manifesto, one principle is to deliver working software frequently from a couple of weeks to a couple of months, with a preference for a shorter timescale (Principles Behind the Agile Manifesto, n.d.). Although the iteration is not time-boxed, this approach allows for producing workable results through continuous improvement across multiple iterations.

Several scholars supported the flexible iteration concept as Ahmed et al. (2018), Larson and Chang (2016) and J. Saltz and Suthrland (2019). Participants P1, P3, and P4 are currently utilising a comparable method because of its flexibility and in need of quick iteration cycles.

The team can establish project milestones in the roadmap to manage expectations and prioritise iterations as they work towards deliverables (J. Saltz et al., 2021). Capability-based iterations offer a more honest approach by acknowledging uncertainty and avoiding false certainty associated with time-boxed iterations (J. Saltz et al., 2021). If the team finishes a task faster than expected, they can quickly move on to the next iteration without the constraints of time-boxed sprints (J. Saltz et al., 2021). This approach aims to enhance adaptability and transparency in the development process.

Developing solutions through flexible iteration helps deliver them faster, even when uncertain. It allows teams to gather early feedback from users, validate assumptions, and make necessary adjustments promptly.

2. Team and Roles

This approach involves a team consisting of a Product Owner, a Project Owner/Manager, a Team Lead, and Project Individual Contributors. These contributors include ML/AI/Data Engineers, Business/Data Analysts, Architects, Software Developers, and UI/UX Engineers. Additionally, stakeholders are involved in the team. Maintaining a small team size is advisable to achieve quick results (Larson & Chang, 2016). This facilitates the active participation of a committed sponsor and fosters close cooperation between the development and user teams (Larson & Chang, 2016).

The Product Owner is the key link between business stakeholders and the development team, ensuring clear communication and aligning project goals with business objectives. AI/ML and DS teams need input from business professionals to work effectively. An intermediary comprehending both domains, such as the product owner, can bridge the gap and foster better understanding between the two parties (Martinez et al., 2021). In their current practice, P1, P2, P3, P4, and P6 all mentioned having a Product Owner. However, P1, P2, P3, and P4 emphasised the importance of the Product Owner understanding business and technical knowledge to be more effective. P6, a data scientist, also acts as the product owner. He stated that having both domain knowledge helps him manage the product efficiently and effectively.

The Project Manager is responsible for various administrative tasks, including resource management, scheduling, and budgeting. They also mediate between business stakeholders and the development team, promoting efficient communication and collaboration. Even though Agile methodologies encourage self-organising teams, the presence of a Project Manager can facilitate coordination, resource management, and project progress monitoring. Based on interviews, having someone accountable for managing resources and timelines in AI/ML projects is favoured to ensure smooth execution (P2, P5, P6).

AI/ML projects require technical expertise and domain knowledge. Team Leads can guide team members and help break down complex tasks, leading to more accurate estimations and efficient task assignments. Someone with more experience could help the team quickly navigate day-to-day issues. The interviews also confirmed this (P2, P4, P6).

Projects involving AI/ML and DS require a diversified team with different skill sets (J. S. Saltz & Grady, 2017) (Hukkelberg & Berntzen, 2019). The crucial roles of ML/AI/Data Engineers, Business/Data Analysts, Architects, Software Developers, and UI/UX Engineers are essential for the project's success as they provide the expertise needed to develop and validate models and solutions. They form the core development team responsible for brainstorming ideas, planning experiments, and executing development activities. Their work produces artefacts that answer business questions.

It is essential to involve stakeholders to ensure that the final product caters to user requirements and adheres to business goals. Literature and interviews emphasise the importance of engaging stakeholders throughout the development process to validate and improve the solution (Qadadeh & Abdallah, 2020) (Ermakova et al., 2021) (Ishikawa & Yoshioka, 2019). Therefore, it was considered them as integral members of the team.

The Agile approach proposes roles prioritising collaboration, communication, and technical expertise in AI/ML projects. These roles align with established Agile principles and practices, essential for addressing challenges of expectation management and business user involvement.

The proposed approach is outlined in Figure 9.

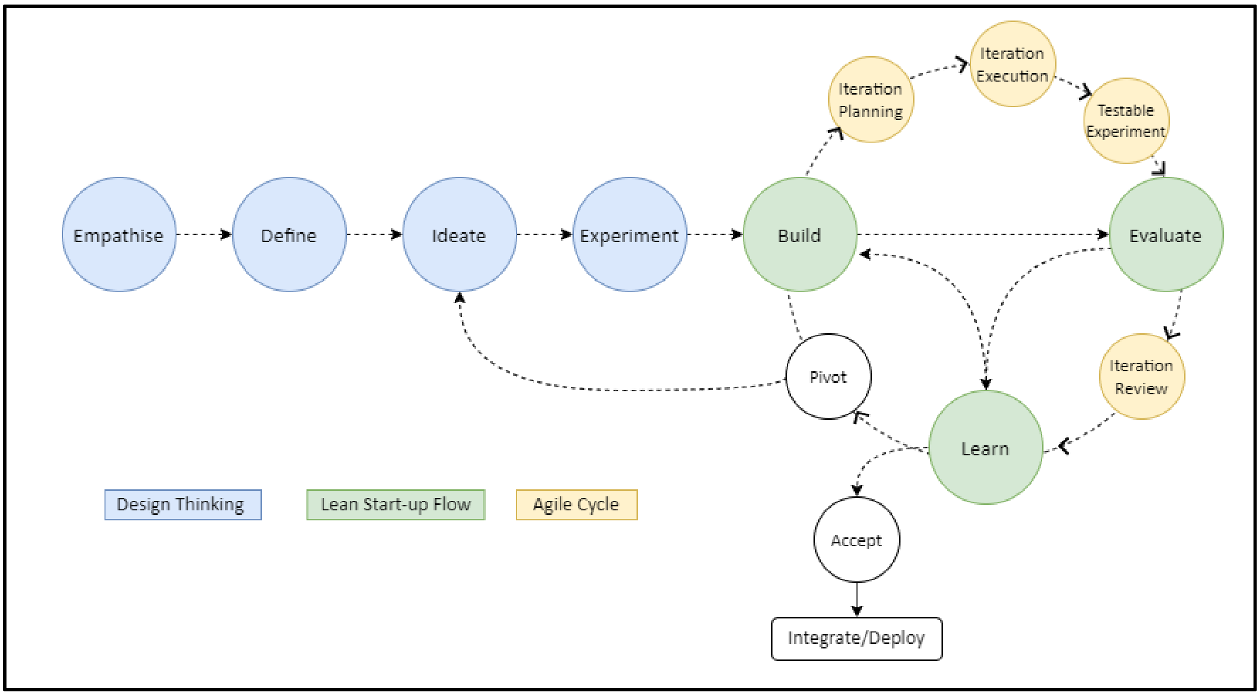

Figure 9 - Overview of the suggested approach

3. Events

- Daily or weekly stand-up meetings

  The team has daily meetings to sync their activities, discuss progress, clear impediments, and plan for the day or week. Teams can decide to hold these meetings either daily or weekly. However, according to the interviews and research, it is suggested that for AI/ML and DS teams, a weekly meeting would be sufficient since there are usually no significant changes that occur daily (Vaidhyanathan et al., 2022) (P1, P2, P3, P4). In order to promote involvement, understanding, and transparency in major decisions, it is important to hold meetings with all team members and key stakeholders (Vaidhyanathan et al., 2022). These meetings facilitate knowledge exchange, improve collaboration, and ensure that every team member is informed of critical decisions made by the team (Vaidhyanathan et al., 2022).

- Iteration Review Meetings

  At these meetings, the team shares their progress during the iteration with stakeholders. This can include codes, models, diagrams, or any other artefact created during the cycle (Larson & Chang, 2016). By regularly sharing project progress and outcomes, the team can validate assumptions and adjust the model based on the feedback. This promotes collaboration and transparency between the development team and business users, ensuring that the delivered solution meets business expectations and user needs.

- Retrospective Meetings

  The team can schedule retrospective meetings every two weeks or monthly to assess their development process, identify improvement areas, and discuss successes and challenges. These meetings create an opportunity for open conversation, enabling ongoing improvements. By working together to solve problems and address issues, the team can optimise their workflow and enhance project outcomes. This approach aligns with the principles outlined in the Agile Manifesto, which states that teams should regularly reflect on their effectiveness and adjust their behaviour accordingly (Principles Behind the Agile Manifesto, n.d.). During interviews, P2, P5, and P6 reported engaging in this practice and finding it a valuable learning experience for the team.

A high-level summary comparison can be found in Table 9 to compare the proposed approach with other frameworks. The frameworks compared are Scrum, Kanban, and CRISP - DM. This summary is based on the work done by J. Saltz and Suthrland (2019)

Aspect	New Approach	Scrum	Kanban	CRISPDM
Iteration	Experiment-based	Time-Based	No	Not Defined
Exploratory Items	Create Tasks as Needed	Not Defined	N/A	Not Defined
Iteration Reviews	After each experiment	After Each Sprint	Not Defined	Not Defined
Iteration Coordination	Kanban Board	Sprint Backlog	Kanban Board	Not Defined
Task Estimation	Use only for PBI Prioritisation	PBI Priority & What Fits into a Sprint	No Task Estimation	Not Defined
Use of PBI	Yes	Yes	Yes	Not Defined
Backlog Selection	When there is the capability to start a new experiment	When Sprint Completes	When there is capacity	Not Defined
Daily Standup	Yes/ Weekly stand-up	Yes	Not Defined	Not Defined
Roles	- Product Owner	- Scrum Master	None Defined	None Defined

	- Project Manager - Team Lead - Project individual contributors - Stakeholders	- Dev Team - Product Owner		

Table 9 - Methodology Comparison

5.2 Solution Evaluation

During the custom Agile methodology development process, two design cycles were conducted. The initial solution was built based on the insights from the first round of interviews and literature reviews. Then, the suggested approach was presented to interview participants to gather more insights and suggestions. These feedback sessions were semi-structured one-on-one interviews; all participants had a positive impression of the work and were willing to share their opinions. Their input was invaluable in helping us fine-tune the methodology, refine its practices, and optimise its alignment with the unique demands of AI/ML and DS projects. This iterative approach was crucial in crafting a methodology that adhered to Agile principles and demonstrated practical applicability in real-world AI/ML and DS development scenarios. The summary of practitioner feedback can be found in Tables 10, 11, 12 and 13.

	First impression of the suggested approach
P2	Most of the practices are already in place in my current organisation. The dynamic sprint concept is a good one. However, sometimes, we work on multiple projects or use cases simultaneously or in the same sprint. When it is not time-bound, then it will become complicated. This method would be best suited when the entire team works on one project. Because if one use case is completed, we could go to the next iteration for another that is not.
P4	I like the fact that you used Kanban as the underlying process. It is not too detailed as you would be focusing on a specific type of project, but it is generalised enough to address many different types of ML projects because there are different types when you say ML projects. I like that you have considered the various types of roles and techniques for collaboration. It is great that you are considering all aspects of the topic. Also, I like this flexibility with task estimation because it is more challenging in ML projects to scope out a project and then create individual tasks.
P5	This idea incorporates elements from various established techniques, resulting in a unique and promising approach. It combines the best aspects of different methods and is a novel combination. Overall, I find it appealing. Of course, there is always room for improvement, but as a starting point, it looks great. Introducing new ways of working can be challenging. If you introduce too many tasks, artefacts, and routines, people may

	become overwhelmed and quit. The key is finding a sweet spot, and this is a good solution.

Table 10 - Evaluation Result Summary: First Impression

	Thoughts on the suggested agile practices and tools, such as Kanban boards and retrospectives.
P2	The Kanban board is really good. We are not recording some sub-tasks that emerged in the sprint because it affects the sprint burndown chart. However, we do a lot of ad-hoc tasks. Those tasks should be recorded for future reference. One advantage of the daily meetings is the constant collaboration with the other people on the call. You understand what is happening around the team or with the others. You could brainstorm and work on some of these calls if someone faces any issues. I like it when we can make it weekly. Because in our last meeting, I remember telling you how much time we spend on calls, which tackles exactly it. I would like to see how the meeting happens if we work with multiple teams, such as the ML, development, and deployment teams because we should be synced once a week.
P4	The Kanban board is very helpful because we could move the task smoothly rather than having a Time-boxed sprint deliverable. That works well with these kinds of projects. That is how we work currently. I like the fact that you consider the different roles as well. It is important to have a product owner or a manager to oversee the project, although we do not have one right now. Having a PO is very helpful when there are cross-functional teams where people work with each other, keeping everyone informed and communicating with stakeholders. That is a good option that you selected to include that kind of leadership, all like management roles in the team to help with estimations, scoping and overseeing the work. Regarding the events, you could combine the iteration demonstration and milestone meetings into a single meeting. I see some overlap with our current context. I like that starting with daily stand-up meetings and transitioning to weekly ones. This is a good idea. Starting with daily stand-ups can help people get accustomed to the process, and then moving to weekly meetings can be more effective for these types of projects where there is a lot of trial and error. Therefore, daily stand-ups may be less beneficial.
P5	Our approach varies depending on the project. For exploratory projects, we use milestones rather than time-bound sprints. However, for projects with a clear scope, we use sprints. That is why it is logical for us to have iterations that are not time-bound. I like the use of the Kanban board, which we currently do.

Table 11 - Evaluation Result Summary: Agile practices and tools

	What potential risks or drawbacks do you foresee in implementing this approach?

P2	We could expect some resistance initially, but I do not see any particular challenges. However, we might need some training on some concepts because I am unaware of some agile-related concepts. As I mentioned earlier, we should look into how this works with teams who work with multiple projects.
P4	I see no challenges right now, as it is easy to adapt. The concepts are self-explanatory. However, it is worth to implement and see the real effect.
P5	I cannot see any drawbacks right now. To see the real effect, we have to implement it and see.

Table 12 - Evaluation Result Summary: Challenges & Drawbacks

	Thoughts on feasibility and effectiveness of implementing in the organisation
P2	This seems easy to implement as it seems lightweight; we could implement this side-to-side with current practice to see the feasibility. Looking at it, it is not hard.
P4	This is easy to adopt way of working. It is feasible to implement this easily. However, we might need some initial training, and PO, PM, and TL can get more sophisticated training.
P5	This is an easy-to-adopt approach, and we do not need extensive training, as more concepts are familiar. It is worth giving a try to implement this.

Table 13 - Evaluation Result Summary: Feasibility of implementation

- **Overall Feeling of the Suggested Approach**

  All three participants (P2, P4, P5) favoured the suggested methodology. While some practices were already in line with their existing processes, they appreciated the dynamic sprint concept. Concerns were raised by P1, managing multiple projects in one iteration, who said this approach might be more suitable for single projects. However, they agreed that the process was promising, combining elements from different techniques freshly and attractively.
- **Agile Practices, Tools & Roles**

  The Kanban board was useful for promoting seamless task flow and flexible project management. Daily stand-up meetings were supported for improving teamwork (P1), but P4 and P5 suggested switching to weekly sessions for better alignment with AI/ML and Data Science projects. P4 acknowledged that involving cross-functional roles like Product Owners (PO) was significant for better communication and estimation.
- **Potential Risks and Drawbacks**

  The only potential issue is resistance to new ideas. All participants said light training would be enough as most concepts are easy to understand.
- **Feasibility and Effectiveness of Implementation**

  According to participants, the methodology was easy to adopt and integrate with existing practices. It was deemed lightweight and familiar, making it feasible for incorporation. Despite requiring initial training, all participants (P2, P4, P5) believed it could be seamlessly incorporated into their organisational context.

- **Changes made upon evaluation.**

  The evaluation results were mostly positive. However, upon reviewing the initial draft of the suggested methodology, setting up milestones and conducting milestone review meetings were included. These milestones were intended to be viewed from a business perspective. However, during feedback from participants P2, P4, and P5, it was mentioned that most of their milestones were technical. P4 even suggested combining the milestone review meeting with the iteration review meeting. This revealed some confusion; the milestone setting and review meeting did not add significant value. Therefore, the decision was made to omit this suggestion in the final solution.

6 Conceptual Solution

6.1 Solution Overview

This chapter will present the ultimate solution to the tailored Agile methodology for AI/ML and DS projects.

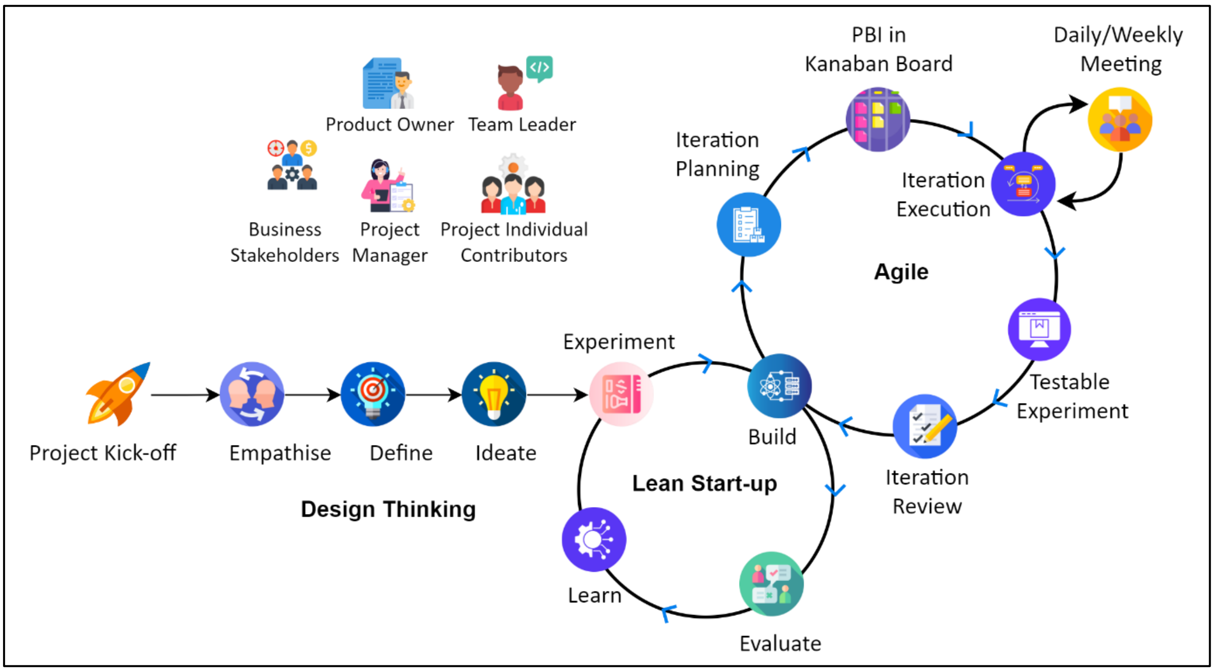

Figure 10 - Final Solution

To summarise, the suggested approach involves merging three methods:

- Kanban: for workflow management
- Design thinking: for comprehending the customer's needs and identifying the business requirement.
- Lean Startup Life Cycle: for refining the model or solution.

Kanban, Lean Startup, and Design Thinking were selected as the foundational principles of the approach due to their compatibility and ability to provide solutions to the key challenges.

Kanban's flow-based approach allows the team to efficiently handle ad hoc tasks and exploratory work. It allows for prioritisation and handling of tasks within the iterations. This flexibility is especially valuable in AI/ML projects where uncertainty often leads to unforeseen tasks and adjustments to the workflow.

Lean Startup practices emphasise frequent engagement with business users and stakeholders, ensuring continuous feedback and validation of assumptions. This iterative approach aligns perfectly with AI/ML projects, where experimentation and hypothesis testing are essential for success. It helps validate the project's direction early on and reduces the risk of investing in the wrong features.

Design Thinking promotes empathy and a deep understanding of user needs, fostering the development of user-centric solutions. This is crucial in AI/ML and DS projects where the end-users needs may evolve rapidly, and a user-driven approach ensures that the final product effectively meets their expectations.

6.1.1 Workflow

1. Project Initiation

In a typical project scenario, after setting up the team, the first activity would be having a kick-off meeting and a workshop to define the project's scope, goals, and success criteria. This team could include roles such as Product Owner, Project Manager, Team Lead, and Project Individual Contributors. At this stage, the team uses Design Thinking techniques to gain insights into the preferences, pain points, and desired features of the target users. This is not only the starting point of the workflow but also the design thinking cycle. The team can use various techniques to thoroughly understand customer needs and expectations, such as conducting interviews and analysing existing data. They should identify the key features and functionalities that can provide value to customers. The team must collaborate with business stakeholders and empathise to understand the real problem. They can move on to the defining and ideation stage as they progress.

As to the next step with the ideation phase, the team and stakeholders can collaborate to create a roadmap that outlines the direction of the entire project. Also, determining which features or use cases should be included in the project. This is based on the project goals and the item's value to the customer. One feature or use case could have multiple experiments. These features use cases, and experiments could add to the project backlog.

2. Backlog Refinement

The stakeholders and development team could pick the first use case or feature depending on the business value and priority. This includes features that directly benefit the customer and tasks or activities that help ensure the project's overall success without providing direct value to the customer. To help prioritise items, the team could use estimations, considering their value and high-level effort. If two items deliver similar value, the team may prioritise a smaller effort item over a larger one.

3. Iteration Planning

When the team is ready to begin a new iteration, they review the prioritised backlog items. They then choose the top item(s) that will form the experiment for the next iteration, focusing on delivering value. The iteration is not strictly time-boxed; However, one iteration should be between 1 week to 4 weeks.

Next, the development team can begin by analysing the work they need to do in the context of statistical and ML techniques. This will lead to the first experiment that helps them choose the best methods to achieve the desired results, such as classification or regression. The chosen analytical approach will affect the requirements, such as data content, formats, and representations. Therefore, the team should inform the stakeholders what to expect and what they need regarding relevant data resources related to the problem domain. The team members can collaborate to plan, delegate tasks, and break down the experiment. The team can enhance collaboration on chosen tasks during iteration planning by adding them to the Kanban board. This will initiate a build cycle within the Lean startup process, allowing the first experiment to commence.

4. Iteration Execution

During the Iteration Execution phase, the team collaborates using the Kanban methodology to work on their chosen experiment. This involves creating a dataset by cleaning and combining data from various sources to make useful variables.

To keep track of the progress, the team visualises their workflow on a Kanban board with columns representing various stages (To do, in progress, done, hold). Furthermore, the team recognises that uncertainties may come with the project. They know that new information may surface, and unforeseen tasks may arise. They can integrate these ad-hoc tasks into their Kanban board and proceed with their work.

Regular communication and collaboration are important, and the team conducts daily or weekly stand-up meetings depending on the team's requirements and agreement between the team and business users.

5. Iteration Demonstration

The team conducts a session to showcase their work to stakeholders and end-users. This progress may not always be a working product but could be any item, such as accuracy metrics, visualisations, or models. They gather feedback and assess the effectiveness of the solution. Using this feedback, the team collects data and insights to develop the next experiment and make any necessary adjustments to the current experiment.

6. Iteration Continuation and Closure

The team can continue working on iterations, concentrating on providing value, improving features, and integrating feedback from stakeholders and end-users. The project will go through several iterations until the product/feature/module achieves a level of maturity that meets the project objectives and customer expectations.

6.1.2 Team & Roles

- Product Owner
  - Act as the bridge between the business stakeholders and the development team.
  - Help with backlog prioritisation.
  - Engage in ongoing analysis by answering questions from the development team.
  - Helps to navigate the impediments.
  - Actively engaging in evaluating the outcome.
- Project Manager
  - Manages the administration tasks such as resources, schedules, and budgets.
  - Liaise between business stakeholders and team.
  - Helps the product owner to navigate the impediments.
- Team Lead
  - Assist with estimating and delegating tasks.
- Project Individual Contributors: ML/AI/Data Engineers, Business/Data Analysts, Architects, Software Developers, UI/UX Engineers
  - Brainstorming ideas and planning experiments
  - Carry out the development activities.
  - Create artefacts to answer the business questions.
  - Provide high-level estimation.
- Stakeholders: Individuals or groups who have a stake in the outcome of the project, such as end-users, business representatives, and other relevant parties, offer feedback and ensure the solution is validated.

6.1.3 Artefacts

- A Product Backlog Item: A Product Backlog Item (PBI) can take various forms, including testable hypotheses, experiments, use cases, features, or tasks.
- Product Backlog: This is a list of the most important features, use cases, and functionalities the product needs. It is the main source of tasks for the team working on the product's development and improvement.

- Kanban Board: This tool uses columns to visually depict the workflow, with each column representing a different stage of development. Teams can adopt standard columns such as To Do, In Progress, Done, Hold, or any other form compatible with the team's desires. It allows the team to keep track of and manage the progress of backlog items and helps set and maintain limits on work in progress (WIP).

6.1.4 Events/Meetings

- Backlog Refinement: The product owner, business representatives, and development team continuously collaborate to discuss, clarify, and prioritise backlog items.
- Iteration Planning: During this meeting, the team will review the prioritised backlog items, choose the next experiment to focus on, and establish the scope and objectives for the upcoming iteration.
- Daily/Weekly Stand-up: The team holds brief daily meetings to synchronise their activities, discuss progress, identify impediments, and plan for the upcoming day or week.
- Iteration Demonstration: The team presents their finished work and features to stakeholders and end-users, collects feedback, and confirms the solution's effectiveness in a session.
- Retrospective: The team will meet after every two iterations to reflect on their work, identify successes and areas for improvement, and adjust in future iterations.

6.2 Practical Implementation – Hypothetical Case Study

Following the comprehensive explanation of a tailored Agile approach for AI, ML, and DS projects, the next step is implementing it in real-world situations. This section examines a case study from "*Machine Learning and Data Analytics for Solving Business Problems: Methods, Applications, and Case Studies"* by Alyoubi et al. (2022) to explore the concrete steps and strategies needed to integrate Agile methodology into actual projects. This will guide organisations combining Agile practices with AI, ML, and DS ventures. Extracting lessons from the case study could reveal how to transform theory into action for a harmonious combination of methodology and practice.

6.2.1 Project Background

Improve sales forecasting for retail point-of-sale.

The retail industry has seen a significant transformation over the past decade with the growth of online retailing. However, the retail supply chain's point-of-sale (PoS) stage remains crucial as customers still value in-store experiences (Udokwu et al., 2022). Retailers are enhancing their shopping environments to increase spending, and technology integration, particularly store management technologies, is driving innovation in retail (Udokwu et al., 2022). This case study explores how data analytics methods, such as machine learning models, can optimise product availability by forecasting demand based on historical data and how they can assist managers in estimating sales and demand to optimise stock levels and availability (Udokwu et al., 2022).

The selected case study is a retail chain that operates hundreds of physical stores in Austria (Udokwu et al., 2022). These stores offer customers point-of-sale (PoS) grocery services (Udokwu et al., 2022). The analysis is based on two years of sales data, from 2017 to 2019 (Udokwu et al., 2022). The focus is on fruit and vegetable products, as they are fast-moving consumer goods that pose challenges for store managers when estimating optimal stock levels and order amounts (Udokwu et al., 2022).

6.2.2 Setting Up the Team

A well-organised team is necessary to implement this project. The project will be implemented in-house with the help of Sales, IT, Data Analytics and Research & Development (R&D) departments. It is recognised that the team requires a diverse team of experts in data science, machine learning, domain knowledge, and technical abilities. The team has been carefully chosen to bring together the best of these fields and includes the following key roles:

- **Product Owner:** Bridge tech and business needs by envisioning a project roadmap, translating business needs to tech, and aligning team efforts with goals.
- **Project Manager (PM):** PM can help plan and manage resources and budget, plan iterations, and coordinate cross-functional efforts to achieve the project goal.
- **Team Lead:** Guides and oversees the team to ensure their work aligns with project and organisational goals.
- **Project Individual Contributors:** Data Scientist, ML Engineer, Software Developers, Quality Assurance Engineer
- **Business Stakeholders and Domain Experts**: Retail Analyst, Sales Manager and Project Steering Committee

6.2.3 Project Kick-off and Workshops

After setting up the team, the first activity in this project scenario would be having a kick-off meeting and a workshop to define the project's scope, goals, and success criteria.

In the kick-off meeting, the team could introduce themselves to each other and business stakeholders could explain the project's main objective and goal. Technical experts briefly explained the nature of the implementing ML project. Clear guidelines have been set for roles, responsibilities, and work methods.

In the next workshop, the team started exploring the business challenge. The team used Design Thinking techniques such as Empathy Mapping, Persona Development, User Journey Mapping, Storyboarding and Rapid Prototypes to gain insights into the preferences, pain points, and desired features of the target users. Below are some examples.

- Empathy Map: Creating empathy maps for store managers and customers to understand their needs, pain points, and motivations. This is useful in tailoring the forecasting solution to address their specific challenges.
- Create Persona: Creating detailed personas for store managers and customers based on research guides the design and development of forecasting models that match user preferences and behaviours.
- Journey Map: Identify the journey and touchpoints of a store manager where sales forecasting could impact decision-making and inventory management. This technique can help uncover opportunities to integrate forecasting insights into their workflow.
- Mind Maps: Use mind maps to organise and connect the different parts of the sales forecasting solution. These could be data sources, model algorithms, user interfaces, and decision points, making it easy to identify any connections or dependencies that could impact the design of the solution.

Moving on to the ideation phase, the team and stakeholders can work together to create a roadmap that charts the course for the entire project. They also decided which features or use cases should be included based on the project goals and their value. It is possible for a single feature or use case to have multiple experiments, and all these additions can be added to the project backlog. Based on the findings, the team has developed the roadmap below.

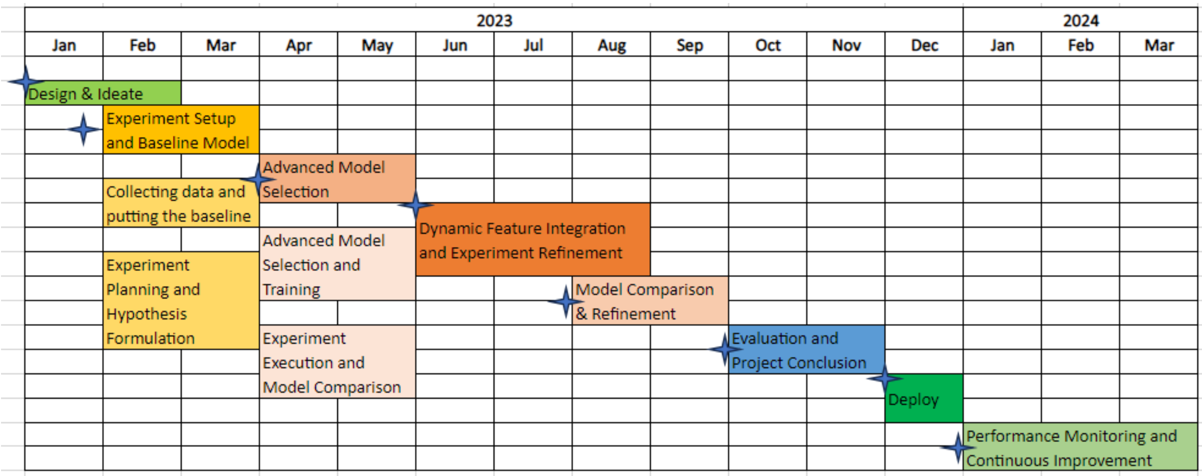

Figure 11 - Project Roadmap

Milestone	Goal	Objective
Experiment Setup and Baseline Model	Data Collection and Baseline Model Implementation	Collect past sales and inventory data for the retail chain's physical shops in Austria and set up a baseline sales forecasting model.
	Experiment Planning and Hypothesis Formulation	Define experiment objectives and hypotheses to drive iterative improvements in sales forecasting.
Advanced Model Development	Advanced Model Selection and Training	Explore and use more advanced ML models to make sales forecasts more accurate.
	Experiment Execution and Model Comparison	Execute experiments to compare the performance of advanced models against the baseline.
Dynamic Feature Integration	Dynamic Feature Integration and Experiment Refinement	Enhance forecasting models by incorporating dynamic features and experimenting with various model configurations.
	Experiment Evaluation and Conclusion	Assess the impact of dynamic feature integration on model accuracy and provide insights for further improvements.
Model Deployment and Performance Monitoring	Model Deployment and Integration	Integrate the optimised sales forecasting model into the retail supply chain for practical use.
	Performance Monitoring and Continuous Improvement	Continuously monitor and improve the deployed model's performance.

Figure 12 - Project Milestones and Goals

6.2.4 Backlog Refinement

Depending on the business value and priority, the business users and development team list the backlog items in the backlog refinement. This refers to end user-oriented features and tasks that contribute to the project's overall success but may not directly impact the customer or sales managers. To help prioritise items, the team used estimations, considering their value and high-level effort. For the starting experiment, the team decided to try to predict retail sales demand by applying classification models and regression models in demand forecasting. The team has created several backlog items displayed in Figure 13.

Backlog Items	Priority
Gather historical sales and inventory data	High
Preprocess and clean data	High
Baseline sales forecasting model - Regressions	High
Baseline sales forecasting model - Classifications	High
Baseline sales forecasting model - Clustering	High
Evaluate baseline model performance	Medium
Review baseline model performance	Medium
Evaluate and select advanced forecasting models	High
Engineer additional features for models	High
Train and validate advanced models	High
Experiment with model configurations	Medium
Compare advanced models against baseline	Medium
Integrate dynamic features into models	High
Experiment with dynamic feature combinations	High
Fine-tune model hyperparameters	Medium
Integrate optimised model into inventory system	High
Develop real-time data feed for model updates	High
Implement model validation and recalibration	Medium
Establish feedback loop for performance monitoring	Medium
Identify areas for continuous improvement	Low

Figure 13 - Project Backlog

6.2.5 Iteration Planning

At the start of the new iteration, the team reviewed the backlog items in order of priority. Based on the review, they select the item(s) that will be the focus of the next iteration to deliver value. In this instance, the team gathered data and pre-processed it to create baseline models. Although the iteration is not strictly time-bound, the team agreed on a 3-week timeframe. The team informed the stakeholders what to expect and what they would need regarding relevant data resources related to the problem domain. The team members delegated tasks and broke down the experiment. During iteration planning, the team added the chosen tasks to the Kanban board (Figure 14). Also, the team agreed to have a weekly update meeting for 15 minutes every Monday and Friday during this three-week iteration. It was planned to have the iteration review meeting on the last Friday of the third week.

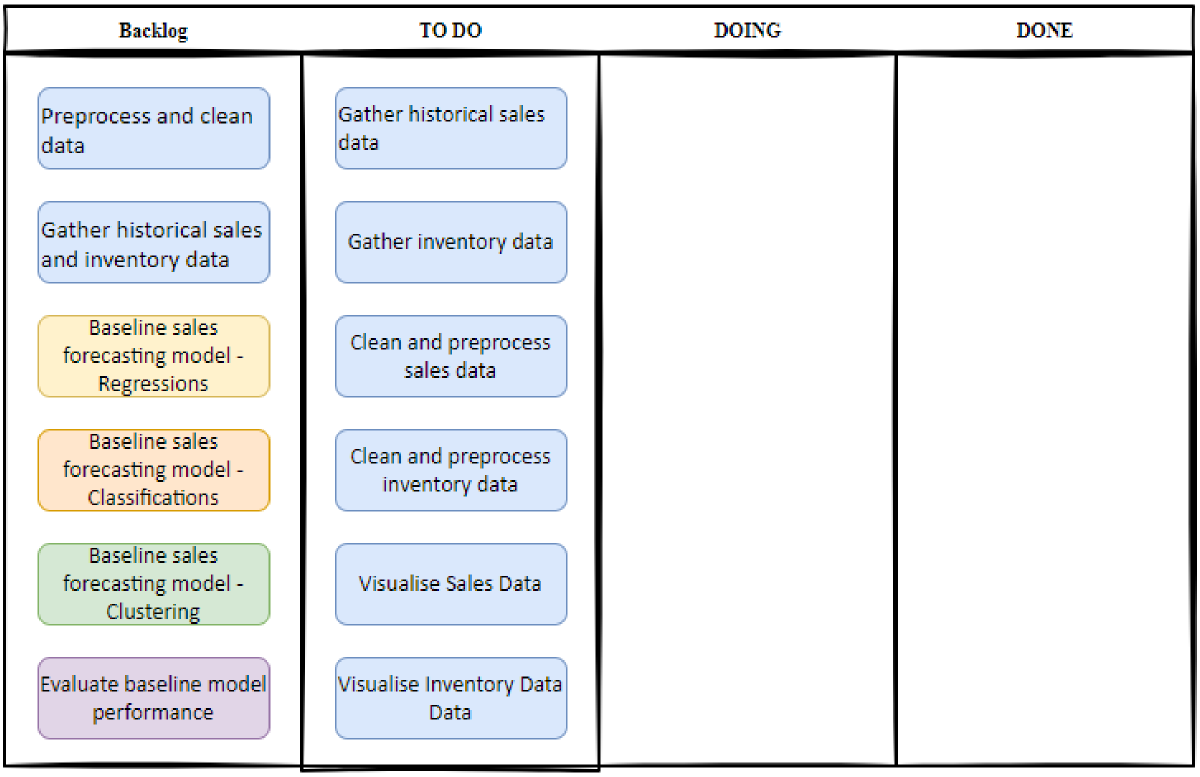

Figure 14 - Team Kanban Board

6.2.6 Iteration Execution

In the Iteration Execution phase, the team worked together to complete their assigned tasks using the Kanban methodology. The team had to collect data from various sources and then clean and combine it to create useful variables for the project.

Traditionally, predicting daily product sales in retail has been viewed as a regression problem (Udokwu et al., 2022). This is because the daily sales quantity for a particular store is represented as a continuous variable (Udokwu et al., 2022). To use a classification algorithm to solve a problem, the input data, like daily sales, needs to be converted into categorical data (Udokwu et al., 2022). The team had to convert the continuous sales data into categories, creating additional tasks. These tasks were added to the Kanban board. The first task was to transform the total daily sales for each product into boxes of products (Udokwu et al., 2022). Then, the product boxes were reduced into categories representing each product's maximum and minimum daily sale boxes (Udokwu et al., 2022). This means that a given category can represent a range of boxes of products (min/max). A total of 12 categories were created to represent the daily product sales (Udokwu et al., 2022).

The team used the Kanban board to visualise their workflow progress. The board has columns that represent different stages, such as "To Do," "In Progress," and "Done". The team was moving their tasks as they progressed. During the weekly stand-up meeting, the new development of categorical data was communicated to the business stakeholders and the product owner; they shared some insights, and their input was valuable in creating the product categories.

6.2.7 Iteration Demonstration

The team had the iteration demonstration after three weeks, as agreed, although the session was to showcase their work to business stakeholders. PM organised this session, and the whole team were presented. The Team Lead recapped the iteration goals, Data Collection and Preprocessing. The team presented the tasks they were working on through visualisations and tables: Last two years' sales and inventory data collection.

As the team had to convert the existing data into categorical data, it was discussed with the business domain experts and validated the categories they had created.

In addition, the team has discovered that predicting sales for PoS retailing with multiple product lists across numerous stores can be challenging. These complexities can negatively affect the accuracy of the forecast predictions. Large PoS retail chains have multiple stores in various demographic locations, each with its own inventory of products (Udokwu et al., 2022). Additionally, the sales patterns of these products vary from store to store. Designing individual models for each product and store is not feasible (Udokwu et al., 2022). Therefore, to improve the accuracy of product sales forecasts, it is essential to develop ML models specific to unique products or groups of stores (Udokwu et al., 2022). This observation was informed to the business stakeholders, which made the domain experts and data analysts analyse and identify possible product categories and store groups. The team said they could use the already preprocessed data for this task. It was decided to have another short iteration to complete this task.

This progress may only sometimes be a working product but could be any item, such as accuracy metrics, visualisations, or models. They gather feedback and assess the effectiveness of the solution. Using this feedback, the team collects data and insights to develop the next experiment and make any necessary adjustments to the current experiment.

The product owner also outlined the upcoming iteration and its objectives. PM and Team Lead discussed the timeline and expectations for the next iteration. The team decided to have short iteration planning to assign the tasks and refine the backlog with new observations.

PM took notes of the meeting details and action items to ensure everyone can refer to them later. It was saved in a folder accessible to the whole team.

6.2.8 Retrospective

After two iteration cycles, PM organised a retrospective meeting. The meeting participants were the PM, Team Lead and Project individual contributors. The team looked back at the goals and objectives of the two completed iterations. During the retrospective, the team highlighted both the accomplishments and challenges faced. The team worked together to devise practical ways to tackle the areas that needed improvement. They assigned specific team members to take responsibility for each step and set deadlines for implementing the suggested changes. Some examples of the improvements they decided on were automating the data collection process and researching more effectively to enhance prediction accuracy in retail stores by combining clustering methods and machine learning forecasting models.

PM recorded the tasks that need to be done and who is responsible for them. This will be helpful for future reference and making sure everyone is accountable.

7 Discussion

This section delves into the key findings and insights from developing and refining the Agile methodology for AI/ML and DS projects. The research questions are addressed by outlining the challenges, requirements, and best practices discovered. Further, potential future research directions will be examined after analysing the study's limitations.

RQ1: What are the unique challenges and requirements of AI /ML and DS projects that are not addressed by existing agile frameworks?

Two significant challenges existing Agile frameworks are compelled to adequately address emerge from investigating the challenges and requirements of AI/ML and DS projects.

1. One of the primary challenges lies in the discrepancy in existing methodologies due to AI /ML and DS projects' very nature of uncertainty and unpredictability. Three challenges fall under this: limited flexibility in the fixed sprint approach, difficulty estimating tasks, and the inability to provide a shippable product at the end of the sprint. Although it was revealed that these challenges occur due to the AI/ML and DS projects' inherited complexity and uncertainty, further analysis showed that these projects' data-driven nature causes the complexity and uncertainty. Compared to conventional software projects such as mobile and web applications, AI/ML and DS project requirements are driven by data. It was impossible to draft functional requirements for AI/ML and DS projects as the functionality they could deliver would depend on the data. Delivery of the solution is only possible if the data is available and the available data are enough to answer the question.

   Further, no universal solutions exist for AI/ML or DS projects. ML engineers and data scientists have to conduct multiple experiments to achieve the final results with the best accuracy, and it takes a considerable amount of time to predict what model would work. Agile frameworks typically prioritise regular delivery and fixed timeframes, but this approach can be challenging for AI/ML and DS projects. Tasks like data cleaning, preparation, and model development often require longer periods to train complex models or handle large datasets, making rapid iteration cycles necessary. Sometimes, ad-hoc tasks arise, making the conventional timeboxing approach less effective.
2. The second major challenge is ensuring stakeholder engagement and transparency. Within this, two specific issues were engaging with business users and managing expectations. Insufficient involvement from business users in AI/ML and DS projects can be attributed to their limited knowledge and understanding of these technologies. A thorough examination of the issue revealed that current methodologies need to establish a clear level of business user involvement, typically confined to providing feedback. This lack of clarity negatively impacts the ability of business users to fully comprehend the project's objectives and how their input can shape its outcomes. As such,

it is imperative to establish a more comprehensive approach that empowers business users to participate more actively in these projects.

As previously stated, AI/ML and DS projects depend on data and may not always result in predetermined solutions. In these cases, data scientists and ML engineers will work to find the best possible answer based on business objectives and available data. To achieve optimal results, technical and business experts must collaborate and understand each other's perspectives. In the current scenario, it is common for business users to only participate in the initial stage of a project rather than follow through with solution exploration.

Business users must embrace the intricacy and unpredictability associated with AI/ML and DS projects. It should be noted that AI/ML solutions are not a panacea that can address every issue, as these solutions are not a one-size-fits-all solution. Research shows that delivering a fully functional AI/ML and DS solution could take up to six months (Ranawana & Karunananda, 2021). Therefore, stakeholders must be involved throughout the project cycle to maintain a realistic understanding of the project's trajectory.

RQ 2: How can an agile methodology be customised better to suit the needs of AI /ML and DS projects?

The present research presents a customised Agile methodology to align Agile practices with AI/ML and DS projects' unique needs. This approach adapts key Agile principles while integrating specific methods tailored to address identified challenges. This solution is influenced by the work of earlier scholars, such as Ahmed et al. (2018) with their lean design thinking methodology (LDTM) for machine learning and modern data projects, and J. Saltz and Sutherland (2019) with their SKI Agile Framework for Data Science. The suggested approach involves integrating three methods: Kanban for workflow management, Design thinking for better business user engagement, and Lean Startup Life Cycle to encourage rapid iterations for refining the model or solution with frequent feedback loops. The 14 table comprehensively summarises how the proposed solution tackles the identified challenges.

Key Challenge	Challenge	Suggested Solution
The discrepancy between current practices and AI/ML/DS Life Cycle Activities	Limited flexibility with time-box approach	The suggested solution prioritises continuous experimentation as opposed to fixed iteration cycles. This approach involves conducting small-scale experiments through trial and error for one to four weeks. Employing a kanban board allows the team to visualise their workflow and seamlessly move tasks across stages as needed, ensuring a flexible approach to managing their work.
	Difficulty in forecasting time for exploration tasks	Recognising that some tasks require more time, the customised approach offers flexible iteration lengths. Smaller experiments use shorter iterations, while more complex experiment development may require multiple iterations. This approach promotes adaptability and eliminates the need for time forecasting during exploration tasks. The Kanban board provides a clear and organised visualisation of the selected experiment and its tasks, allowing the

		team to easily identify and prioritise the most important tasks based on their value and significance.
	Not having a shippable product by the end of a sprint	Using the Lean Startup cycle and experiment-driven iteration (which involves building, evaluating, and learning), each iteration will produce a testable single experiment. This experiment may yield various outputs, such as codes, models, diagrams, or other artefacts created during the cycle (Larson & Chang, 2016).
Stakeholder Engagement and Transparency	Business User Engagement	This approach places importance on involving domain experts throughout the development process. Lean Start-Up feedback cycles (Build - Measure - Learn) and Design Thinking principles are used for this. In the project initiation phase, the technical team can use design thinking tools to understand the domain and empathise with business users. Similarly, business users can use technical experts to grasp the problem and its dynamics. After the ideation and trying out the experiment, the results will be shared with business users to assess the solution's effectiveness. Both teams can use these learnings to improve or pivot the solution.
	Expectation Management	Similarly to the above, utilising Design Thinking can foster empathy and a comprehensive understanding of user needs and concerns. Involving stakeholders throughout the process and setting realistic expectations during project initiation and each iteration can enhance alignment with user expectations and reduce unexpected issues. Lean Startup feedback loops aid in maintaining communication with business users and stakeholders, providing them with a realistic view of progress. These feedback loops include Iteration Planning, Daily/Weekly Stand-up meetings, and Iteration Reviews. Furthermore, the Kanban board enhances transparency on ongoing work. Furthermore, these efforts are supported by various roles, including Product Owner, Project Manager, Team Leader, and individual contributors. Business users are also considered an integral part of the project team.

Table 14 - Customised Approach Overview

RQ3: What are the best practices for implementing a customised agile methodology in AI/ML and DS projects?

As a result of this research, several effective approaches have been uncovered that can facilitate the seamless integration of a tailored Agile methodology into AI/ML and Data Science projects. These findings influenced the customised approach.

1. **Encourage Collaboration between Business Users and AI/ML and DS Project Team:** It is crucial to achieve synergy and guarantee project success by facilitating efficient communication and collaboration among data scientists, machine learning engineers, business domain experts, and project managers. Creating detailed plans for AI/ML and DS projects can be difficult when limited high-level planning information is available. However, collaboration can help overcome this challenge by defining expectations and improving stakeholder communication.

2. **Adopt rapid, short cycle and flexible iterations:** When engaged in AI/ML and DS projects, the data involved is often unstructured, necessitating an interactive and iterative analysis to uncover insights. To accelerate this process, it is recommended to adopt a "short-cycle Agile" approach, characterised by more flexible and faster sprints (Larson & Chang, 2016). In addition, the iterations should be amenable to ad-hoc tasks. At the same time, the planning approach must remain flexible enough to accommodate changing project dynamics because AI/ML and DS projects require ongoing modifications to achieve optimal results.

3. **Include specialised roles:** To achieve success in AI/ML and DS projects, it is crucial to have specialised roles like data scientists, machine learning engineers, or domain experts (Martinez et al., 2021). These roles can be either internal team members or external collaborators. However, team leads, managers, and data scientists/ML engineers may have varying opinions on the value of these roles. Integrating new roles into a team requires reflection on diverse aspects and perspectives of value (Hukkelberg & Berntzen, 2019). It is essential to manage expectations and establish a shared understanding of value to ensure successful collaboration within the team and across the organisation (Hukkelberg & Berntzen, 2019).

4. **Prioritise experimentation and prototyping:** In AI/ML and DS projects, it is customary to engage in experimentation and prototyping of various algorithms, models, and approaches. It is advisable to allocate specific time or columns dedicated to experimentation as part of the process (Ishikawa & Yoshioka, 2019). This practice facilitates the validation and exploration of different concepts. For data scientists and ML engineers to perform optimally, they must be allowed to experiment and explore the data (Hukkelberg & Berntzen, 2019). A rigid work environment can impede their ability to be effective. Moreover, data scientists require a high degree of autonomy in their work (Hukkelberg & Berntzen, 2019).

5. **Incorporate feedback loops**: In AI/ML projects, it is helpful to incorporate feedback loops into the process. Gathering input from stakeholders allows for necessary adjustments. Assessing customer readiness and project viability is important, and halting or cancelling a project may be necessary. Sometimes, creating meaningful business value for customers should be prioritised over technical performance (Vial et al., 2022).

7.1 Limitations

Although this customised Agile methodology is promising for AI/ML and Data Science projects, it is essential to recognise its limitations. This chapter evaluated the constraints we faced during this research, which could occur in design science research (Larsen et al., 2020; Dresch et al., 2014). These limitations may affect the generalisability and dependability of the results.

Construct Validity: One limitation is that AI/ML and DS projects are often very complicated due to their research nature and differ from standard software development projects. Despite the efforts to describe this research's key ideas and structures, it takes much work to get a complete picture because different organisations have different practices and maturities. Finding a good balance between agile practices and their applications in AI/ML and DS projects might have led to oversimplification or loss of complexity in some areas.

Representation Validity: The study's use of a small sample size for interviews to establish the problem definition and feedback sessions could undermine the statistical validity of the findings. Due to the qualitative nature of the semi-structured interviews may not fully represent the diversity of perspectives and experiences within the larger AI/ML and Data Science communities. Increasing the sample size and diversity could enhance the precision and dependability of the findings.

Challenges in Research Implementation: Time constraints limited the creation and evaluation of the customised Agile methodology. This led to the execution of iterative design cycles and gathering feedback within a specific timeframe. Consequently, the depth of data collection and methodology refinement may have been compromised. In-depth exploration of nuances and additional iterative enhancements could be aided by additional research requiring more time.

External Validity: Due to time constraints and practicalities, it was not possible to implement this methodology in real-world project teams; therefore, it is impossible to estimate the amount of time and money it could save. Additionally, it was not tested if the methodology's effectiveness may vary based on project scale, complexity, domain expertise, and resource availability. Although the customised Agile methodology was created for AI/ML and Data Science projects, it may not suit all projects and organisational contexts. It may require additional validation and customisation to ensure applicability in various contexts. Additional research with diverse organisations and project sizes could enhance its adaptability and efficacy.

Data Input Validities: Using semi-structured interviews as the primary method for collecting qualitative data may have led to subjectivity and potential bias in interpreting participant responses. Lack of additional data acquisition methods, such as surveys or observational analysis, may also limit the breadth and depth of understanding. Combining various data collection forms could provide a more comprehensive comprehension of the participants' perspectives.

The evaluation outcomes of this study mainly revolve around the feedback and opinions of the participants regarding the tailored Agile methodology. However, the study only reviewed the literature to compare its findings with other Agile frameworks or project management methodologies. It would be beneficial to compare this methodology with other approaches to gain a more comprehensive understanding of its advantages and disadvantages.

7.2 Future Work

In order to improve the credibility of this research, the next step should be to expand the sample size and gather feedback from real-world project scenarios. This could help refine the problem definition and strengthen the research's findings.

In terms of future research, it would be valuable to explore the scalability and applicability of the customised Agile framework across various AI/ML and Data Science projects. Investigating the impact of this methodology on project outcomes, team collaboration, and stakeholder satisfaction could provide valuable insights. Comparative studies between the customised Agile approach and other project management methodologies could also shed light on this methodology's unique advantages and drawbacks.

While conducting this study, it was discovered that some challenges faced in AI/ML and DS projects using Agile methodologies overlap with those faced in conventional software projects. However, the causes of these challenges are different due to the inherent differences between the two types of projects. Further research could explore why these challenges differ from conventional Agile software development, as there were insufficient studies.

Lastly, this study assumed that a customised methodology could be built upon the Agile methodology. While interviews and literature supported this, there is an opportunity to conduct a more open methodology combining multiple methodologies.

8 Conclusion

This research project aimed to create a customised Agile methodology for AI/ML and DS projects. Although Agile methodologies have become popular in software development, adapting them to AI/ML and DS projects is difficult due to their unique characteristics, such as complexity, the requirement for multiple experiments, and rapid iteration. Although other frameworks are available, combining various software development methodologies within an organisation can be challenging. Consequently, other scholars put in various efforts to address this need. Our study was conducted to overcome these obstacles by developing a tailored Agile methodology that caters to the needs of AI/ML and DS projects.

Using the Design Science Research (DSR) strategy, this research involved conducting semi-structured interviews and a comprehensive literature review to build the problem definition and identify the requirements. Six practitioners were interviewed, providing valuable insights into using Agile methodologies for AI/ML and DS projects. The literature review further enhanced our understanding of the challenges and recommendations mentioned during these interviews.

We have identified two inherent core challenges in AI/ML and DS projects through these efforts. The first key challenge was that the conventional Agile paradigms needed to be revised with these projects' uncertainties, requiring a flexible approach to iterations. Three main issues need to be addressed when it comes to this key challenge:

I. The fixed sprint approach could be more flexible, which can limit progress.
II. It can be challenging to estimate tasks accurately.
III. Delivering a shippable product at the end of a sprint is often impossible.

After the analysis, it became clear that these challenges are mainly due to the complexity and uncertainty of AI/ML and DS applications. However, further analysis revealed that the data-driven nature is the key reason that causes complexity and uncertainty.

The second key challenge is the importance of stakeholder engagement and transparency. The interview participants and existing literature have highlighted the need for continual involvement of business users and domain experts throughout the Agile lifecycle.

To tackle these challenges, it was suggested to use an Agile methodology tailored to AI/ML and DS project needs. This methodology combines Kanban, Design Thinking, and Lean Startup Life Cycle principles. Using this combination, it is possible to effectively deal with task flexibility, uncertainties, and stakeholder communication.

After formulating the Agile methodology tailored to the unique requirements and challenges of AI/ML and DS projects, the suggested approach was shared with the interview participants. In the second round of interviews, three participants evaluated and validated proposed enhancements. All participants agreed that while this approach is not entirely new, it innovatively combines the most effective methods from existing methodologies. Its lightweight structure makes it promising for easy implementation.

This study was conducted to help with Agile methodologies for AI/ML and DS projects. It examines the difficulties when conventional Agile methods are applied to AI/ML and DS projects. For further research, exploring the practical applications and effectiveness of the proposed methodology in different organisational contexts was suggested.

9 References

Abrahamsson, P., Oza, N., & Siponen, M. T. (2010). Agile Software Development Methods: A Comparative Review1. In *Springer eBooks* (pp. 31–59). https://doi.org/10.1007/978-3-642-12575-1_3

Ahmad, M., Dennehy, D., Conboy, K., & Oivo, M. (2018). Kanban in software engineering: A systematic mapping study. *Journal of Systems and Software*, *137*, 96–113. https://doi.org/10.1016/j.jss.2017.11.045

Ahmed, B., Dannhauser, T., & Philip, N. (2018). A lean design thinking methodology (LDTM) for machine learning and modern data projects. *Computer Science and Electronic Engineering (CEEC)*, 11–14. https://doi.org/10.1109/ceec.2018.8674234

Akif, R., & Majeed, H. (2012). Issues and challenges in scrum implementation. *International Journal of Scientific & Engineering Research*, *3*(8).

Alqudah, M., & Razali, R. (2016). A Review of Scaling Agile Methods in Large Software Development. *International Journal on Advanced Science, Engineering and Information Technology*, *6*(6), 828. https://doi.org/10.18517/ijaseit.6.6.1374

Alyoubi, B., Ncir, C. B., Alharbi, I., & Jarboui, A. (2022). *Machine learning and data analytics for solving business problems: Methods, Applications, and Case Studies*. Springer Nature.

Alzubi, J. A., Nayyar, A., & Kumar, A. (2018). Machine Learning from Theory to Algorithms: An Overview. *Journal of Physics*, *1142*, 012012. https://doi.org/10.1088/1742-6596/1142/1/012012

Amershi, S., Begel, A., Bird, C., DeLine, R., Gall, H., Kamar, E., Nagappan, N., Nushi, B., & Zimmermann, T. (2019). Software Engineering for Machine Learning: A Case Study. *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*. https://doi.org/10.1109/icse-seip.2019.00042

Ameta, U., Patel, M., & Rathore, N. S. (2023). Fusing Artificial Intelligence with Scrum Framework. *Social Science Research Network*. https://doi.org/10.2139/ssrn.4428286

Atwal, H. (2019). *Practical DataOps: Delivering Agile Data Science at Scale*. Apress.

Azevedo, A., & Santos, M. F. (2008). KDD, SEMMA AND CRISP-DM: A PARALLEL OVERVIEW. *IADIS European Conference Data Mining 2008 (Part of MCCSIS 2008)*, 182–185.

Batra, D. (2017). Adapting Agile Practices for Data Warehousing, Business Intelligence, and Analytics. *Journal of Database Management*, *28*(4), 1–23. https://doi.org/10.4018/jdm.2017100101

Blomster, M., & Koivumäki, T. (2021). Exploring the resources, competencies, and capabilities needed for successful machine learning projects in digital marketing. *Information Systems and e-Business Management*, *20*(1), 123–169. https://doi.org/10.1007/s10257-021-00547-y

Bortolini, R. F., Cortimiglia, M. N., De Moura Ferreira Danilevicz, Â., & Ghezzi, A. (2018). Lean Startup: a comprehensive historical review. *Management Decision*, *59*(8), 1765–1783. https://doi.org/10.1108/md-07-2017-0663

Brasjo, C., & Lindovsky, M. (2019). *Machine learning project management - A study of project requirements and process in early adoption* [MA Thesis]. Chalmers University of Technology.

Brocke, J. V., Hevner, A. R., & Maedche, A. (2020). Introduction to Design Science Research. In *Progress in IS* (pp. 1–13). https://doi.org/10.1007/978-3-030-46781-4_1

Carter, E., & Hurst, M. (2019). *Agile Machine learning: Effective Machine Learning Inspired by the Agile Manifesto*. Apress.

Davenport, T. H. (2018). From analytics to artificial intelligence. *Journal of Business Analytics*, *1*(2), 73–80. https://doi.org/10.1080/2573234x.2018.1543535

De Sá Araújo, C. N., Santos, I. M., Canedo, E. D., & Araujo, A. P. F. (2019). Design Thinking Versus Design Sprint: A Comparative Study. In *Lecture Notes in Computer Science* (pp. 291–306). Springer Science+Business Media. https://doi.org/10.1007/978-3-030-23570-3_22

Dikert, K., Paasivaara, M., & Lassenius, C. (2016). Challenges and success factors for large-scale agile transformations: A systematic literature review. *Journal of Systems and Software, 119*, 87–108. https://doi.org/10.1016/j.jss.2016.06.013

Dingsøyr, T., Nerur, S. P., Balijepally, V., & Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*, *85*(6), 1213–1221. https://doi.org/10.1016/j.jss.2012.02.033

Dobrigkeit, F., De Paula, D., & Uflacker, M. (2018). InnoDev: a software development methodology integrating design thinking, scrum and lean startup. In *Understanding innovation* (pp. 199–227). Springer International Publishing. https://doi.org/10.1007/978-3-319-97082-0_11

Dresch, A., Lacerda, D. P., & Antunes, J. a. V., Jr. (2014). *Design science research: A Method for Science and Technology Advancement*. Springer.

Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information & Software Technology*, *50*(9–10), 833–859. https://doi.org/10.1016/j.infsof.2008.01.006

Ermakova, T., Blume, J., Fabian, B., Fomenko, E., Berlin, M., & Hauswirth, M. (2021). Beyond the Hype: Why Do Data-Driven Projects Fail? *Proceedings of the Annual Hawaii International Conference on System Sciences*. https://doi.org/10.24251/hicss.2021.619

Fayyad, U. (1997). Data mining and knowledge discovery in databases: Implications for scientific databases. *Proceedings. Ninth International Conference on Scientific and Statistical Database Management (Cat. No.97TB100150)*. https://doi.org/10.1109/ssdm.1997.621141

Fitriani, W. R., Rahayu, P., & Sensuse, D. I. (2016). *Challenges in agile software development: A systematic literature review*. https://doi.org/10.1109/icacsis.2016.7872736

Flora, H., & Chande, S. (2014). A systematic study on agile software development methodologies and practices. *International Journal of Computer Science and Information Technologies*, *5*(3), 3626–3637.

Franková, P., Drahošová, M., & Balco, P. (2016). Agile Project Management Approach and its Use in Big Data Management. *Procedia Computer Science, 83*, 576–583. https://doi.org/10.1016/j.procs.2016.04.272

Grady, N. W., Payne, J., & Parker, H. (2017). Agile big data analytics: AnalyticsOps for data science. *International Conference on Big Data*. https://doi.org/10.1109/bigdata.2017.8258187

Hesenius, M., Schwenzfeier, N., Meyer, O., Koop, W., & Gruhn, V. (2019). *Towards a Software Engineering Process for Developing Data-Driven Applications*. https://doi.org/10.1109/raise.2019.00014

Hevner, A. R., & Chatterjee, S. (2010). Design Science research in information systems. In *Integrated series on information systems* (pp. 9–22). https://doi.org/10.1007/978-1-4419-5653-8_2

Hevner, March, Park, & Ram. (2004). Design Science in Information Systems Research. *MIS Quarterly*, *28*(1), 75. https://doi.org/10.2307/25148625

Hoda, R. (2019). *Agile Processes in Software Engineering and Extreme Programming – Workshops: XP 2019 Workshops, Montréal, QC, Canada, May 21–25, 2019, Proceedings*. Springer Nature.

Hron, M., & Obwegeser, N. (2018). *Scrum in Practice: an Overview of Scrum Adaptations*. https://doi.org/10.24251/hicss.2018.679

Hukkelberg, I., & Berntzen, M. N. (2019). Exploring the Challenges of Integrating Data Science Roles in Agile Autonomous Teams. *Lecture Notes in Business Information Processing*, 37–45. https://doi.org/10.1007/978-3-030-30126-2_5

Identifying and Addressing 6 Key Questions when Using Data Driven Scrum. (2021, December 15). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9671930&casa_token=d8hnG_5VtlYAAAAA:3-yqHqgyLidUd5c5CLqaGmiSSOdxspmHUPWDiBqmox3vUqELVW4oIr1IJKB8GuoYQl9-B9hvZvEqFg&tag=1

Ishikawa, F., & Yoshioka, N. (2019). *How Do Engineers Perceive Difficulties in Engineering of Machine-Learning Systems? - Questionnaire Survey*. https://doi.org/10.1109/cesser-ip.2019.00009

Jurney, R. (2013). *Agile Data Science: Building Data Analytics Applications with Hadoop*. “O’Reilly Media, Inc.”

Jurney, R. (2017). *Agile Data Science 2.0: Building Full-Stack Data Analytics Applications with Spark*. “O’Reilly Media, Inc.”

Kirovska, N., & Saso, K. (2015). Usage of Kanban methodology at software development teams. *Journal of Applied Economics and Business 3.3*, 25–34.

Koehnemann, H. (2023, March 13). *Design Thinking - Scaled Agile framework*. Scaled Agile Framework. https://scaledagileframework.com/design-thinking/

Kohl, P., Schmidts, O., Klöser, L., Werth, H., Kraft, B., & Zündorf, A. (2021). STAMP 4 NLP – An Agile Framework for Rapid Quality-Driven NLP Applications Development. *Communications in Computer and Information Science*, 156–166. https://doi.org/10.1007/978-3-030-85347-1_12

Krasteva, I., & Ilieva, S. (2020). Adopting agile software development methodologies in big data projects: A systematic literature review of experience reports. *2020 IEEE International Conference on Big Data (Big Data)*, 2028–2033. https://doi.org/10.1109/BigData50022.2020.9378118

Larsen, K. R., Lukyanenko, R., Mueller, R. M., Storey, V. C., VanderMeer, D., Parsons, J., & Hovorka, D. S. (2020). Validity in design Science research. In *Lecture Notes in Computer Science* (pp. 272–282). https://doi.org/10.1007/978-3-030-64823-7_25

Larson, D., & Chang, V. (2016). A review and future direction of agile, business intelligence, analytics and data science. *International Journal of Information Management*, *36*(5), 700–710. https://doi.org/10.1016/j.ijinfomgt.2016.04.013

Leau, Y. B., Loo, W. K., Tham, W. F., & Tan, S. Y. (2012). Software development life cycle AGILE vs traditional approaches. *2012 International Conference on Information and Network Technology (ICINT 2012)*, *37*, 1.

Lwakatare, L. E., Raj, A., Bosch, J., Olsson, H. H., & Crnkovic, I. (2019). A Taxonomy of Software Engineering Challenges for Machine Learning Systems: An Empirical Investigation. *Lecture Notes in Business Information Processing*, 227–243. https://doi.org/10.1007/978-3-030-19034-7_14

Martinez, I., Viles, E., & Olaizola, I. G. (2021). Data Science Methodologies: Current Challenges and Future Approaches. *Big Data Research*, *24*, 100183. https://doi.org/10.1016/j.bdr.2020.100183

Matharu, G. S., Mishra, A., Singh, H., & Upadhyay, P. (2015). Empirical Study of Agile Software Development Methodologies. *ACM Sigsoft Software Engineering Notes*, *40*(1), 1–6. https://doi.org/10.1145/2693208.2693233

McKinsey Technology Trends Outlook 2022. (2022, August 24). McKinsey & Company. https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/the-top-trends-in-tech

Mishra, A. K. (2021). AI4R2R (AI for Rock to Revenue): A review of the applications of AI in mineral processing. *Minerals*, *11*(10), 1118. https://doi.org/10.3390/min11101118

Najdawi, A., & Shaheen, A. (2021). Which Project Management Methodology is better for AI-Transformation and Innovation Projects? *2021 International Conference on Innovative Practices in Technology and Management (ICIPTM)*. https://doi.org/10.1109/iciptm52218.2021.9388357

Nascimento, E., Nguyen-Duc, A., Sundbø, I., & Conte, T. (2020). Software engineering for artificial intelligence and machine learning software: A systematic literature review. *Journal of System and Software*. https://doi.org/10.48550/arXiv.2011.03751

Ongsulee, P. (2017). Artificial intelligence, machine learning and deep learning. *International Conference on ICT and Knowledge Engineering*. https://doi.org/10.1109/ictke.2017.8259629

Polyzotis, N., Roy, S., Whang, S. E., & Zinkevich, M. (2017). Data Management Challenges in Production Machine Learning. In *International Conference on Management of Data*. https://doi.org/10.1145/3035918.3054782

Principles behind the Agile Manifesto. (n.d.). https://agilemanifesto.org/principles.html

Putta, A., Paasivaara, M., & Lassenius, C. (2018). Benefits and Challenges of Adopting the Scaled Agile Framework (SAFe): Preliminary Results from a Multivocal Literature Review. In *Lecture Notes in Computer Science* (pp. 334–351). Springer Science+Business Media. https://doi.org/10.1007/978-3-030-03673-7_24

Qadadeh, W., & Abdallah, S. (2020). An Improved Agile Framework For Implementing Data Science Initiatives in the Government. *2020 3rd International Conference on Information and Computer Technologies (ICICT)*. https://doi.org/10.1109/icict50521.2020.00012

Ranawana, R., & Karunananda, A. S. (2021). An Agile Software Development Life Cycle Model for Machine Learning Application Development. *2021 5th SLAAI International Conference on Artificial Intelligence (SLAAI-ICAI)*. https://doi.org/10.1109/slaai-icai54477.2021.9664736

Raschka, S., Patterson, J. T., & Nolet, C. J. (2020). Machine Learning in Python: Main Developments and Technology Trends in Data Science, Machine Learning, and Artificial Intelligence. *Information*, *11*(4), 193. https://doi.org/10.3390/info11040193

Rising, L. W., & Janoff, N. (2000). The Scrum software development process for small teams. *IEEE Software*, *17*(4), 26–32. https://doi.org/10.1109/52.854065

Rupali, M., & Amit, P. (2017). A Review Paper on General Concepts of "Artificial Intelligence and Machine Learning." *International Advanced Research Journal in Science, Engineering and Technology*, *4*(4), 79–82. https://doi.org/10.17148/iarjset/nciarcse.2017.22

Saltz, J. S. (2015). The need for new processes, methodologies and tools to support big data teams and improve big data project effectiveness. *International Conference on Big Data*. https://doi.org/10.1109/bigdata.2015.7363988

Saltz, J. S., & Grady, N. W. (2017). The ambiguity of data science team roles and the need for a data science workforce framework. *2017 IEEE International Conference on Big Data*. https://doi.org/10.1109/bigdata.2017.8258190

Saltz, J. S., Shamshurin, I., & Crowston, K. (2017). Comparing data science project management methodologies via a controlled experiment. *Proceedings of the . . . Annual Hawaii International Conference on System Sciences*. https://doi.org/10.24251/hicss.2017.120

Saltz, J. S., Sutherland, A., & Jombart, T. (2021). Identifying and Addressing 6 Key Questions when Using Data Driven Scrum. *2021 IEEE International Conference on Big Data (Big Data)*. https://doi.org/10.1109/bigdata52589.2021.9671930

Saltz, J., & Suthrland, A. (2019). SKI: An Agile Framework for Data Science. *2019 IEEE International Conference on Big Data (Big Data)*. https://doi.org/10.1109/bigdata47090.2019.9005591

Schön, E. M., Escalona, M., & Thomaschewski, J. (2015). Agile Values and Their Implementation in Practice. *International Journal of Interactive Multimedia and Artificial Intelligence*, *3*(5), 61. https://doi.org/10.9781/ijimai.2015.358

Schröer, C., Kruse, F., & Gómez, J. M. (2021). A Systematic Literature Review on Applying CRISP-DM Process Model. *Procedia Computer Science*, *181*, 526–534. https://doi.org/10.1016/j.procs.2021.01.199

Serban, A., Van Der Blom, K., Hoos, H., & Visser, J. (2020). Adoption and Effects of Software Engineering Best Practices in Machine Learning. *Proceedings of the 14th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. https://doi.org/10.1145/3382494.3410681

Shahapurkar, S. (2016). Crossing the chasm: Deploying machine learning analytics in dynamic real-world scenarios. *Arizona State University ProQuest Dissertations Publishing*.

Sharma, S., & Hasteer, N. (2016). *A comprehensive study on state of Scrum development*. https://doi.org/10.1109/ccaa.2016.7813837

Singla, K., Bose, J., & Naik, C. (2018). Analysis of Software Engineering for Agile Machine Learning Projects. *2018 15th IEEE India Council International Conference (INDICON)*. https://doi.org/10.1109/indicon45594.2018.8987154

Tashtoush, Y. M., Darweesh, D. O., Hussari, G., Darwish, O., Darwish, Y., Issa, L. B., & Ashqar, H. I. (2022). Agile Approaches for Cybersecurity Systems, IoT and Intelligent Transportation. *DOAJ (DOAJ: Directory of Open Access Journals)*. https://doi.org/10.1109/access.2021.3136861

Turetken, O., Stojanov, I., & Trienekens, J. J. M. (2017). Assessing the adoption level of scaled agile development: a maturity model for Scaled Agile Framework. *Journal of Software*, *29*(6), e1796. https://doi.org/10.1002/smr.1796

Uchida, K., Adachi, H., Kikkawa, T., Kirihara, A., Ishida, M., Yorozu, S., Maekawa, S., & Saitoh, E. (2016). Corrections to "Thermoelectric Generation Based on Spin Seebeck Effects" [DOI: 10.1109/JPROC.2016.2535167]. *Proceedings of the IEEE*, *104*(7), 1499. https://doi.org/10.1109/jproc.2016.2577478

Udokwu, C., Brandtner, P., Darbanian, F., & Falatouri, T. (2022). Chapter 4: Improving sales prediction for Point-of-Sale retail using machine learning and clustering. In *Machine Learning and Data Analytics for Solving Business Problems - Methods, Applications, and Case Studies* (pp. 55–74). Springer Nature Switzerland AG. https://doi.org/10.1007/978-3-031-18483-3

Uysal, M. P. (2022). Machine learning and data science project management from an agile perspective: Methods and challenges : In contemporary challenges for agile project management. *IGI Global*, 73–88. https://doi.org/10.4018/978-1-7998-7872-8.ch005

Vaidhyanathan, K., Chandran, A., Muccini, H., & Roy, R. (2022). Agile4MLS—Leveraging agile practices for developing machine Learning-Enabled systems: An industrial experience. *IEEE Software*, *39*(6), 43–50. https://doi.org/10.1109/ms.2022.3195432

Vial, G., Cameron, A., Giannelia, T., & Jiang, J. (2022). Managing artificial intelligence projects: Key insights from an AI consulting firm. *Information Systems Journal*. https://doi.org/10.1111/isj.12420

Wan, Z., Xia, X., Lo, D., & Murphy, G. C. (2021). How does Machine Learning Change Software Development Practices? *IEEE Transactions on Software Engineering*, 1. https://doi.org/10.1109/tse.2019.2937083

Westenberger, J., Schuler, K., & Schlegel, D. (2022). Failure of AI projects: understanding the critical factors. *Procedia Computer Science*, *196*, 69–76. https://doi.org/10.1016/j.procs.2021.11.074

What is agile? (2023, March 27). McKinsey & Company. https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-is-agile

What is design thinking & why is it important? | HBS Online. (2022, January 18). Business Insights Blog. https://online.hbs.edu/blog/post/what-is-design-thinking

10 Appendix 1 – Interview Protocol – Interview Round 1

1. Introduction
   i. Greet the participant and introduce yourself.
   ii. Explain the purpose of the interview, which is to discuss developing a customized agile framework for AI/ML and DS projects.
2. Practicalities
   i. Consent - Video On
   ii. Consent - Recording
   iii. Provide an overview of the interview process, including estimated duration and any confidentiality measures.
3. Disclaimers
   i. Explain the confidentiality measures in place and assure the participants that their responses will be anonymised, and no names of participants or organisations will be shared and strictly used for research purposes only.
   ii. The interviewee has the right to withdraw their participation without any consequences.
4. Background Information
   i. Introduction about Interviewee
5. Questions
   i. Could you briefly describe the type of AI/ML or Data Science project you are working on?
      i. How is the project being executed, i.e. what is the development process or methodology being used?
      ii. What is your role or contribution to the project?
   ii. What are the top 3 challenges teams typically face when working on AI and ML projects using Agile methodologies?
   iii. How did the process look like to overcome them?
   iv. In your view, is there any need for new ways of communication and collaboration among team members with diverse skill sets and areas of expertise in AI and ML projects?
      i. What existing communication and collaboration methods are used in AI/ML projects?
      ii. What are the limitations of current communication and collaboration methods in AI/ML projects?
   v. Do you think new roles are needed to better meet team members' different communication and collaboration requirements with diverse skills and areas of expertise in AI and ML projects?
      i. How do existing AI/ML project team roles contribute to communication and collaboration?

ii. What are the limitations of existing roles in addressing the communication and collaboration requirements of team members with diverse skills and areas of expertise in AI/ML projects?

iii. What potential new roles could be introduced to better meet these requirements?

vi. Based on your experience, do you think a customized Agile framework is necessary for AI and Machine Learning projects? Why or why not?

vii. What are the top 3 key areas we should focus on when developing a customized Agile Framework?

viii. What are the top 3 key metrics that you think would be useful for measuring the Agile framework's success in improving project outcomes, team productivity, and stakeholder satisfaction?

ix. What future developments or trends do you foresee in Agile methodologies for AI and Machine Learning projects?

x. Do you want to share anything that you might think is important to me but did not cover in the questions?

6. Feedback and Iteration – Next steps
7. Conclusion

i. Thank the participants for their time and input.

11 Appendix 2 – Interview Protocol – Interview Round 2

Interview Protocol - Round 2 – Evaluation and Feedback

1. Introduction
   i. Greet the interviewee.
   ii. Explain the purpose of the interview- Discuss developing a customised agile methodology for AI/ML and DS projects and get feedback.
2. Practicalities
   i. Video On - If the interviewee is not comfortable with the video, it can be turned off
   ii. Check Consent for recording as per the purpose of analysis
   iii. Give an overview of the interview process, estimated duration
3. Disclaimers
   i. Explain the confidentiality measures in place and assure the participants that their responses will be anonymised and that no names of participants or organizations will be shared and strictly used for research purposes only.
   ii. The interviewee has the right to withdraw their participation without any consequences.
4. Background Information – Optional as this was done in round 1
5. Proposed Changes and Modifications
   i. Present the suggested changes and modifications to the Agile methodology for AI /ML and DS projects – Optional and if required only.
   ii. Ask the participant if they have questions or clarifications - Optional and, if required, only.
6. Evaluation and Feedback Questions
   i. Looking at this model, what do you recognise and what does make more sense? Or does it not make sense?
   ii. What are your thoughts on the suggested agile practices and tools, such as Kanban boards and retrospectives?
   iii. What potential risks or drawbacks do you foresee in implementing this approach? Are there any specific areas that might need further refinement or consideration?
   iv. How feasible and effective do you believe this approach would be in your context or organisation?
   v. What training or support might be necessary to ensure team members can effectively use the Agile methodology for AI and ML projects?
7. Additional Questions and Wrap-up
8. Thank the participants for their valuable input and contributions to the research project
9. Confirm any next steps or follow-up actions, if applicable.