LEAN AGILE DESIGN: A SOFTWARE DEVELOPMENT APPROACH

by

Amol Madankumar Kate, MBA, ME

DISSERTATION

Presented to the Swiss School of Business and Management Geneva

In Partial Fulfillment

Of the Requirements

For the Degree

DOCTOR OF BUSINESS ADMINISTRATION

SWISS SCHOOL OF BUSINESS AND MANAGEMENT GENEVA

NOVEMBER 2025

LEAN AGILE DESIGN: A SOFTWARE DEVELOPMENT APPROACH

by

Amol Madankumar Kate

APPROVED BY

Dissertation chair

RECEIVED/APPROVED BY:

Renee Goldstein Osmic

Admissions Director

Dedication

<Optional. If you do not include a dedication, delete this text. **Do not** delete the section break that follows this text. Press Ctrl+Shift+8 to see the section break.>

Acknowledgements

It gives me an immense pleasure in writing this acknowledgement toward the submission of my DBA thesis. This is a very important and transformative milestone in my life. I would like to express my sincere appreciation and gratitude to my Mentor Dr. Anuja Shukla, whose continuous support and extended help could help me manage my extensive studies along with my job and family responsibilities. Without her guidance and direction, I could not have finished my research.

I would like to thank to Prof. …….., Dissertation chair and Prof. ……., Admission Director for providing me this wonderful opportunity to conduct cutting-edge research on such a new generation subject.

This would also not have been possible without the blessings of my parents. I would also like to thank my Spouse Mrs. Priya Kate and my Son Aaradhy for their, patience and continuous encouragement during this journey.

I also thank to all the professionals who have taken out their time to fill the survey form, which was an important input my research. I also extend my appreciation to all my teachers, mentors, university staff, and friends for their continuous support and help whenever needed.

And last but definitely not the least, thanks to god almighty for inspiring me to conduct this wonderful, fulfilling and intellectually satisfying research.

Amol Kate

ABSTRACT

LEAN AGILE DESIGN: A SOFTWARE DEVELOPMENT APPROACH

Amol Madankumar Kate
2025

Dissertation Chair: <Chair’s Name>
Co-Chair: <If applicable. Co-Chair’s Name>

This research analyses the current pitfalls observed in complex software product development projects in industry from in-depth review of literature by using Systematic Integrated Mapping Method (SIMM). We analyzed various development frameworks, integrated models and real industry project data such as functional efforts, schedule variance and root causes. Consequently, lists of dependent (Y, yi) and independent variables (xi) were defines, a questionnaire was developed and survey was conducted to capture voices of practitioners from software development projects. We analyzed the resultant dataset using various statistical, ANOVA, Regression models and Advanced Machine Learning models to develop Project Success equation, predictive models to predict project success for variable inputs and generate a list of significant variables which impacts project success. An advanced mathematical model was built to included effect of non-linearity, variable interaction effects and Saturation. Based on these critical insights, using Deep-Thinking and Progressive Model Development technique, we developed a

software product development project framework – Lean Agile Design (LAD), which is a cohesive meta-model using principles from Design Thinking, System Thinking, Agile and Lean Start-up. Overall, a research-to-practice continuum is created, which contributes to the body of knowledge of product development, software engineering and project management. The models built during the course of the research are scalable and applicable to industrial and academic scenarios to enhance efficiency, predictability and on-time launch of high-quality products.

Keywords: Design Thinking, Agile, Lean, System Thinking, Software Product Development, Project Management, Machine Learning

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I:

INTRODUCTION

1.1 Introduction

There has been a remarkable development in innovation and technology in the 21st century. With the advent of Industry 5.0, there is a race to develop increasingly complex products, which are sustainable, customer-centric and can make people's lives better (Xu, 2021). Few of the prominent examples of these technologies are observed in the field of software product development, Artificial Intelligence, cyber-physical product development and tech-startups. With future industrial revolution, Industry 6.0, more complex technologies such as quantum computing, nanotechnology and blockchain will be leveraged to increase efficiency and effectiveness of the products built in preceding industrial revolution phase (Chourasia et. al., 2022).

In the modern business world, innovation is not just a luxury but a necessity for existence in the market. Further, the addition of ultra-complex product development practices such as software product development adds complexity to the business processes. To ensure the on-time high-quality software product launch which satisfies human needs, a singular product development approach will not sail the boat of the program, but a multi-faceted approach is the need of the hour.

To navigate the complexity of the product development process, multi-process approach has been used in which, more than one method is used throughout the product development lifecycle. The integration between various philosophies such as Design Thinking (DT), Agile Methodologies (AM), and Lean Principles has fascinated various researchers, both industry practitioners and scholars. Individually each of the philosophies is functioning as used in industry now a days, however together these three approaches form a powerful triad which enhances the efficiency of software product development process. The combined approach also fuels innovation and redefines the way of organizational transformation and long-term success.

The report is organized into six sections such as Introduction, Literature review, Methodology, Results, Discussion, and Summary, implication & recommendation.

1.2 Research Problem

It is very complex to build an original product consisting multiple technologies from ideation, and launch it predictably. How might we build a better and flawless process to tackle this problem? The proposition is that, combining different philosophies by leveraging best features from each of them and integrate them seamlessly will help to manage complex programs in a better manner.

The main research theme is: How can different philosophies/ methodologies be integrated to build a better product development process?

This takes us to the formation of Research Questions (RQ) as follows:

- RQ1: What contemporary Design thinking, Lean thinking and Agile methodologies are practiced in software development Industries?
- RQ2: What are currently available 'integrated models' developed by researchers?
- RQ3: What are pitfalls and corresponding governing variables in current ultra-complex product development programs?
- RQ4: What is the relationship between various independent variables and program success?
- RQ5: How various philosophies can be integrated to build up a model to eliminate the common pitfalls?

- RQ6: How can the conceptual integrated model be deployed in innovation ecosystem?

The RQ1 is a research component which shows what methodologies on DT, agile and lean are practically followed by academicians and industry professionals. RQ2 is about conducting a detailed & systematic literature review and analyzing various integrated models developed by various authors. RQ3 is about finding the insights on challenges in currently used product development methods (individual or integrated models) and defining factors which impacts the project success. In RQ4, we will try to analyze and quantify the impact of the variables on the project success. RQ5 is about building a model using various approaches such as design thinking, agile methodologies, and lean principles to arrest common issues in product development. The RQ6 is about the analysis and recommendations on model application in industry.

1.3 Purpose of Research

The ultimate purpose of this research work is to develop an integrated model deploying best practices from various philosophies such as Design thinking, Agile methodologies, Lean principles etc. for managing complex programs such as software product development.

In light of this bigger objective, we have defined few sub objectives such as:

► Study the currently used methods/ models in design thinking, lean and agile by industry practitioners and academicians. We will develop a novel review literature review method for conducting this research.

► Explore various integrated models developed by researchers for software project management and derive various themes of model development.

► Conduct detailed literature review, survey and project data analysis to come up with a list of pitfalls which will drive the development of model.

► Identify various factors responsible for project success, establish relationship between these factors and project delivery parameters, and build up predictive models to assess project success for given input project parameters.

► Generate ideas on integrating different philosophies to develop a model which can overcome the pitfalls in managing ultra-complex programs.

► Find out ways to deploy the model in innovation ecosystem and provide recommendations on how the model can be used in in software programs.

1.4 Significance of the Study

New products are the outcomes of projects! There has been a tremendous shift in human mindset from classical waterfall approaches, which were once considered gold standard in project management, to contemporary Agile deployment. Agile methodologies are revolutionized (Rigby, 2016) from PDCA to agile manifesto & scrum to Scaled agile framework (SAFe) & DevOps to make them leaner, faster, and predictable. Even after the great revolution in agile and waterfall methodologies, numerous pitfalls are observed while using these methodologies for ultra-complex system development.

The current program methodologies work well for simple programs, however may not for ultra-complex projects considering the fragmented nature of current methodologies. Few representative ultra-complex programs are mentioned in figure 1.1 below.

Figure 1.1 Representative Examples of Ultra-complex Programs

Based on the study of models developed by various authors, it is observed that, there is insufficient work done to improve existing models and to create integrated models for software product development programs. So, as a part of the research we intend to build a conceptual model integrating various principles from Design thinking, Agile methodologies, Lean thinking etc.

This research will help in better understanding of the various philosophies associated with agile project management for product delivery, design thinking for better understanding of customer requirements and lean principles for higher productivity in development process. The findings from this research will definitely add value to the knowledge base of software project engineering and management. The results of this research will help the industry practitioners and academicians in their day-to-day project work as they understand the most important aspects of the projects, use the novel framework and can predict the performance of the project to have on-target product delivery.

CHAPTER II:

REVIEW OF LITERATURE

2.1 Methodology of Literature review

The objective of this literature review is to study the existing research work conducted on integrated models using principles from various methodologies such as Design Thinking (DT), Agile Methodologies (AM) and Lean Start-up (LS). Various models are critically analyzed and categorized into different themes. A dichotomy has been established between models to present their strengths and potential improvement areas.

A literature review framework – "Systematic Integrated Mapping Method" – is developed (figure 2.1) which is largely a combination of Systematic Mapping Studies (Petersen et. al., 2008; Parizi et. al., 2022), Integrative Literature Review (Torraco, 2005; Snyder, 2019) and benchmarking of other literature studies.

Snyder (2019) proposed three methodologies for conducting literature review such as Systematic, Semi-systematic and Integrative based on the purposes such as critique or synthesize, width of research question and search strategy. Integrating literature review method helps in researching and integrating various philosophies to build a novel model (Torraco, 2005).

Petersen et. al. (2008) has recommended a very effective approach to conduct Systematic Mapping Studies (SMS) inspired from the Object-Oriented Mapping Studies (Bailey, 2007). Da Silva et. al. (2011) and Brhal et. al. (2015) have used SMS method for conducting literature review for user-centered agile software development.

Parizi et. al. (2022) and Pereira et. al. (2018) have effectively implemented the SMS method for systematic literature review of DT application in agile development.

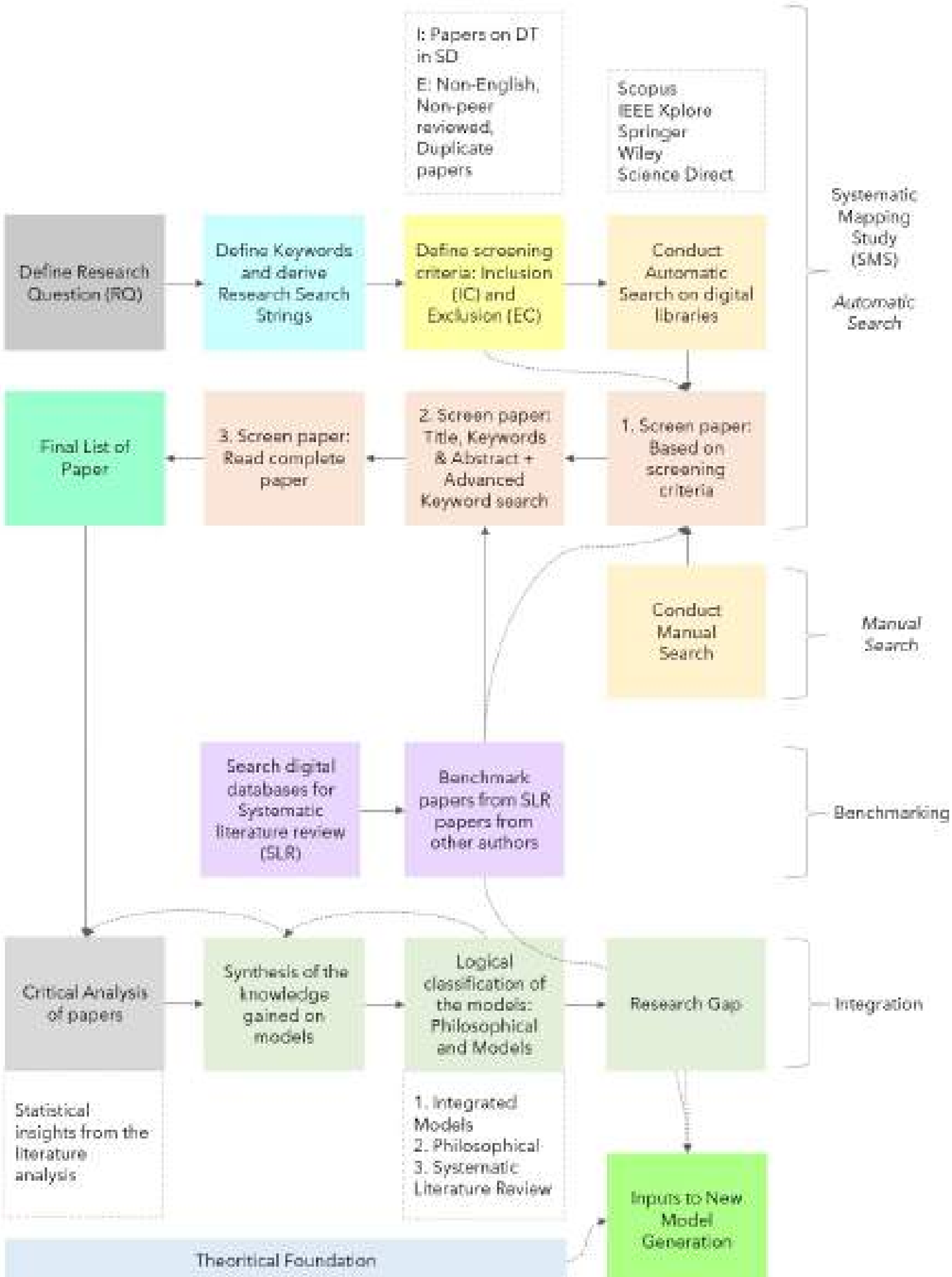


Figure 2.1 Literature Review Methodology – Systematic Integrated Mapping Method

We have leveraged these methods to build our literature review method - "Systematic Integrated Mapping method" - for the purpose of this research work (figure 2.1). The step-by-step literature review is conducted using Systematic Integrated Mapping Method (SIMM) model in the upcoming sections.

2.2 Research Search String

Based on the various keywords and their types, search string elements are developed as shown in Table 2.1 below:

Table 2.1 Keywords and associated Search String Elements

Type	Keywords	Search String Element
Design Thinking	Design Thinking	"design think*"
	User Centered Design	X
	Human Centered Design	X
Software	Software Development	"software development"
	Software Engineering	"software engineering"
	Software Process	"software process"
	Software Project	"software project"
	Software Project Management	"software project management"
	Software Program	"software program"
	Software Program Management	"software program management"
Agile	Agile Methodology	X
	Agile Development	X
Lean	Lean	X
	Lean Principles	X
	Lean Management	X
	Lean Startup	X

We have excluded the Human centered design and user centered design, as they are included in the design thinking search string. Also, all the search string elements with respect to Agile and Lean are also excluded as they will narrow down the search result eliminating useful papers associated with DT application in software development.

All search string elements are connected using Boolean operators such as “AND” and “OR”, using truncation symbols such as wildcard character “*”. Accordingly, the keyword the search string is defined as follows:

("design think") AND (("software development") OR ("software engineering") OR ("software process") OR ("software project") OR ("software project management") OR ("software program") OR ("software program management"))*

Figure 2.2 Research Search String

2.3 Screening Criteria

Various screening criteria are defined as shown in Table 2.2 below.

Table 2.2 Inclusion and Exclusion criteria

Screening	Code	Screening Criteria
Inclusion	I1	All papers (Research & Conference) on application DT in software development as per research search string.
Exclusion	E1	Duplicate papers and books
	E2	Papers which are not in English language
	E3	Papers which are not peer reviewed
	E4	Papers which do not meet the research search string criteria

2.4 The Search Result

The automatic search in various digital libraries has resulted in a list of numerous papers. The search is conducted for the period from year 2000 to 2023. Table 2.3 depicts the number of papers as a search result in each database. The various digital libraries used for sourcing the paper are based on the context of the research. The libraries such as Scopus, IEEE, Springer, Wiley, Science Direct are suitable for research work in design thinking, Software product development, agile project management, system thinking and Lean start-up.

Table 2.3 The search result from digital library

#	Digital library	Number of search results
1	Scopus	252
2	IEEE Xplore	88
3	Springer Link	1269
4	Wiley	176
5	Science Direct	485
6	Manual Search	23
	Total	**2293**

2.5 Literature Funneling – Screening 1 based on Exclusion criteria

All the papers are screened based on the inclusion and exclusion criteria. Out of 2293 papers, total 124 papers (5.4%) were eliminated based on the exclusion criteria such as duplicate papers (E1, non-English language (E2), & non-peer reviewed (E3) papers. Table 2.4 shows the summary of the first screening of papers.

Table 2.4 Paper Screening based on Inclusion and Exclusion criteria

#	Digital library	Number of search results	Exclusion (E1 + E2 + E3)	Result After Screening 1
1	Scopus	252	14	238
2	IEEE Xplore	88	39	49
3	Springer Link	1269	62	1207
4	Wiley	176	7	169
5	Science Direct	485	2	483
6	Manual Search	23	0	23
	Total	**2293**	**124**	**2169**

2.6 Preliminary Analysis of Papers – Screening 2 based on abstract, key words & title

The papers are analyzed and filtered based on the abstract, keywords and titles. 86.7% of papers were filtered during the second screening resulting in 182 papers for full reading as shown in table 2.5.

Table 2.5 Paper Screening based on abstract, keywords and title

#	Digital library	Result After Screening 1	Excluded in Screening 2	Result After Screening 2
1	Scopus	238	146	92
2	IEEE Xplore	49	33	16
3	Springer Link	1207	1170	37
4	Wiley	169	162	7
5	Science Direct	483	476	7
6	Manual Search	23	0	23
	Total	**2169**	**1987**	**182**

2.7 Detailed Analysis and Classification – Screening 3 based on full reading of papers

A total of 182 papers, after the second screening, were fully read and studied. Based on the study, 108 papers, i.e., 4.7% were excluded in third screening as shown in table 2.6, which resulted in the final list of 74 papers for inclusion in literature review.

Table 2.6 Paper Screening based on full paper reading

#	**Digital library**	**Number of search results**	**Result After Screening 1**	**Result After Screening 2**	**Excluded in Screening 3**	**Final Paper for Research**
1	Scopus	252	238	92	64	28
2	IEEE Xplore	88	49	16	14	2
3	Springer Link	1269	1207	37	19	18
4	Wiley	176	169	7	6	1
5	Science Direct	485	483	7	5	2
6	Manual Search	23	23	23	0	23
	Total	**2293**	**2169**	**182**	**108**	**74**

The 74 papers were reread and studied in depth, then analyzed and classified according to type such as Integrated models, Philosophical, and Systematic literature review. The classification of the final list of papers is shown in table 2.7 and figure 2.3. Integrated models, which are highly useful for the research work to critically analyze and study are 44 nos. The Philosophical papers do not explicitly share a model, but talks about generic methodologies and framework, which can add inputs to our research. Finally, the 6 Systematic Literature Review papers are useful for understanding approach taken by various authors to conduct systematic review of literature for this topic and also helps to understand the kind of research already done in summarized form.

Table 2.7 Paper analysis and classification

#	Digital library	Integrated Models	Philosophical	Systematic Literature Review	Total
1	Scopus	18	7	3	28
2	IEEE Xplore	-	2	-	2
3	Springer Link	8	7	3	18
4	Wiley	-	1	-	1
5	Science Direct	2	-	-	2
6	Manual Search	16	7	-	23
	Total	**44**	**24**	**6**	**74**

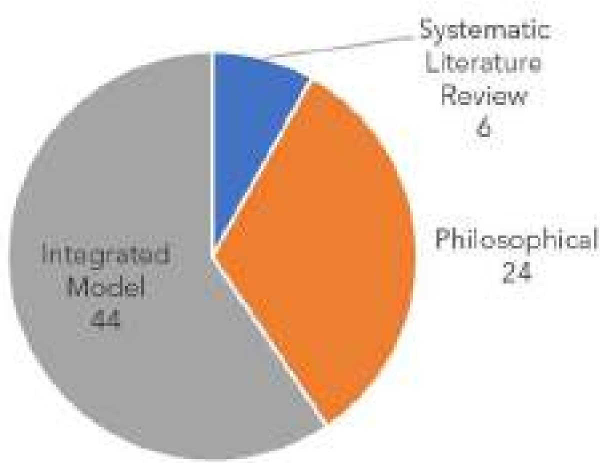


Figure 2.3 Type of Research paper (Total: 74 papers)

2.8 Summary of Literature review funneling

Figure 2.4 below shows the summary of paper filtering at each stage of SIMM. 74 papers i.e., 3.2% of the total search of 2293 papers were found useful for literature review and further research and analysis work.

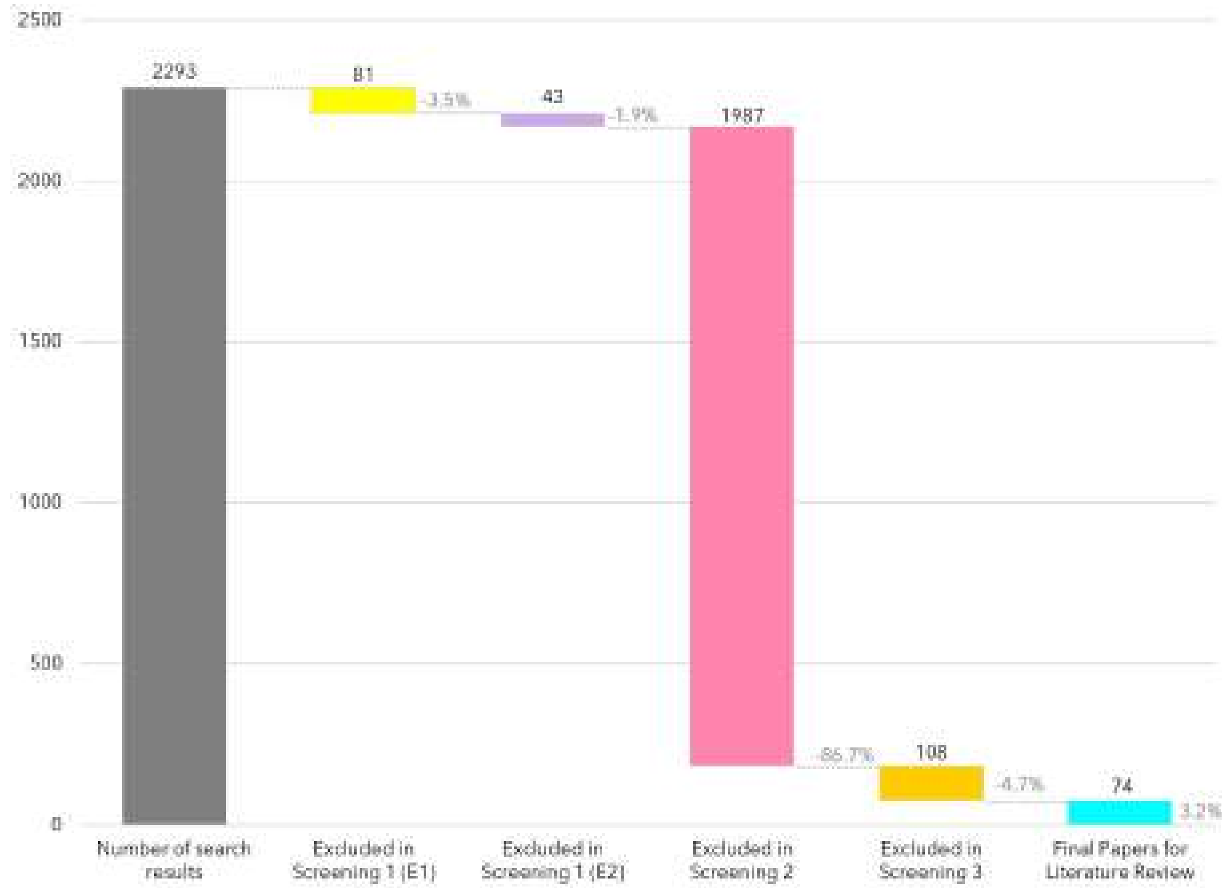


Figure 2.4 Progressing literature funneling with SIMM

During literature research, various Design Thinking methods are observed. Table 2.8 indicates various design thinking models along with their phases

Table 2.8 Design Thinking Method from literature research

#	DT Model	Number of Phases	Phase
1	DCIDT	2	Divergent - Convergent
2	Brown	3	Inspiration - Ideation - Implementation
3	Souza & Silva Model	3	Immersion - Ideation - Prototyping
4	Double Diamond	4	Discover - Define - Develop - Deliver
5	Dunn & Martin	4	Abduction (Generate Ideas) - Deduction (Predict Consequences) - Test - Induction (Generalize)
6	Glomann Model	4	Research- Ideation - Prototyping - Evaluation
7	IBM DT	4	Understand - Explore - Prototype - Evaluate
8	ISO 9241-210:2019	4	Understand context - Specify user requirements - Produce a solution - Evaluate design
9	Luchs	4	Discover - Define - Create - Evaluate
10	D.School	5	Empathize - Define - Ideate - Prototype - Test
11	Mainel & Leifer	5	Define/ Redefine - Need finding & Synthesis - Ideate - Prototype - Test
12	mDT (multidisciplinary Design Thinking)	5	Use model - System model- Interaction model - Model consolidation - System development
13	Google Design Sprint	6	Understand - Define - Sketch - Decide - Prototype - Validate
14	Grossman-Kahn and Rosensweig Model	6	Define the challenge - Observe people - Form Insights - Frame Opportunities - Brainstorm ideas - Try experiments
15	Hasso-Plattner Institute (HPI)	6	Understand - Observe - Define Point of view - Ideate - Prototype - Test
16	IDEO Human-Centered Design (HCD)	6	Observation - Ideation - Rapid prototyping - User feedback - Iteration - Implementation
17	Sandino & Matey	7	Define - Explore - Ideate - Prototype - Select - Implement - Review
18	Hiremath & Sathiyam Model	7	Scoping - 360 degree Research - Synthesis - Ideation - Prototyping - Validation - Implementation

It is observed that D.School design thinking is the most used method followed by ISO 9241-210:2019 and Hasso-Plattner Institute (HPI) methods (figure 2.5).

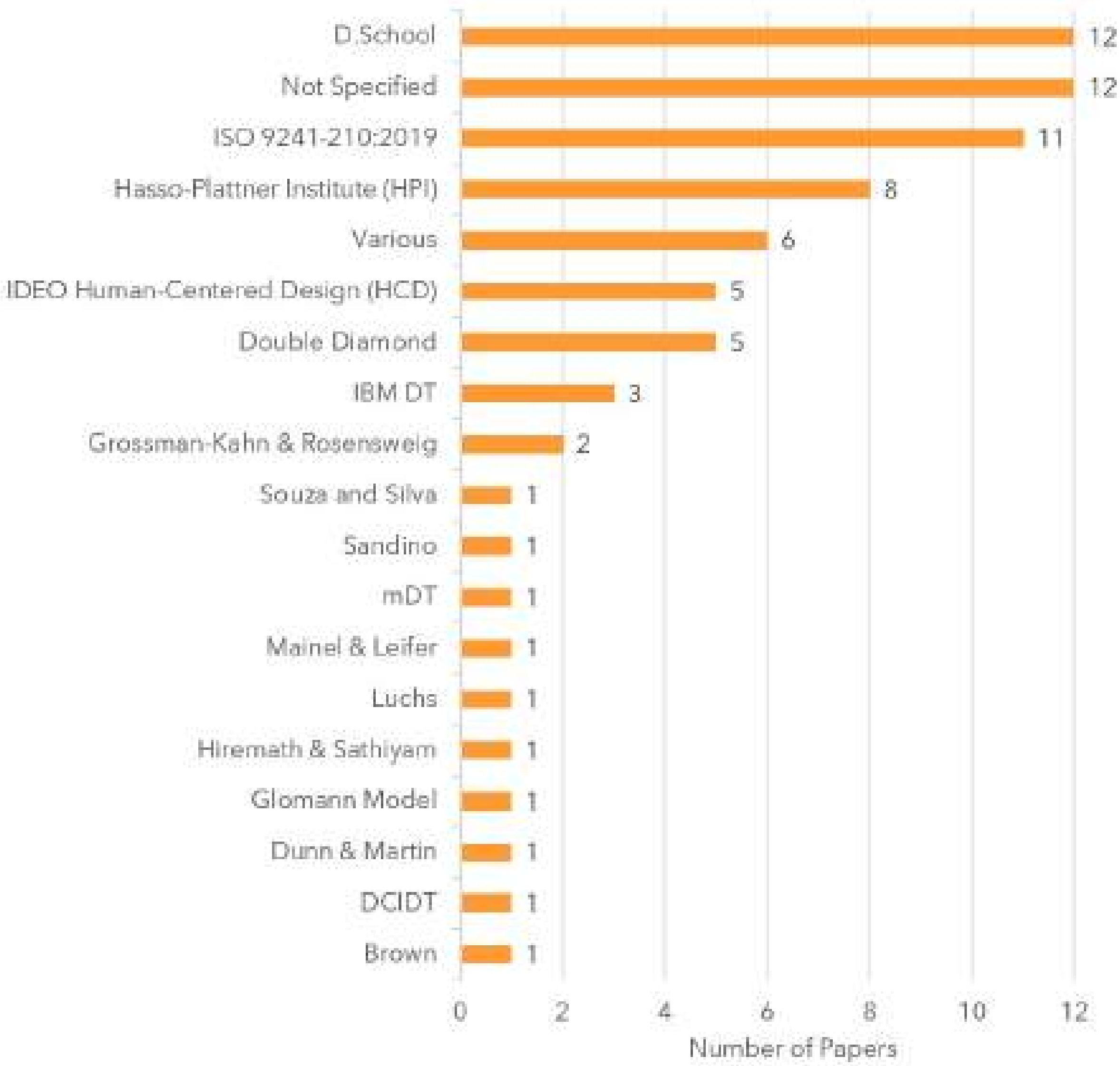


Figure 2.5 Distribution Design Thinking Methods

Scrum is found to be the most popular method amongst researchers for model integration followed by Extreme programming (XP) (figure 2.6).

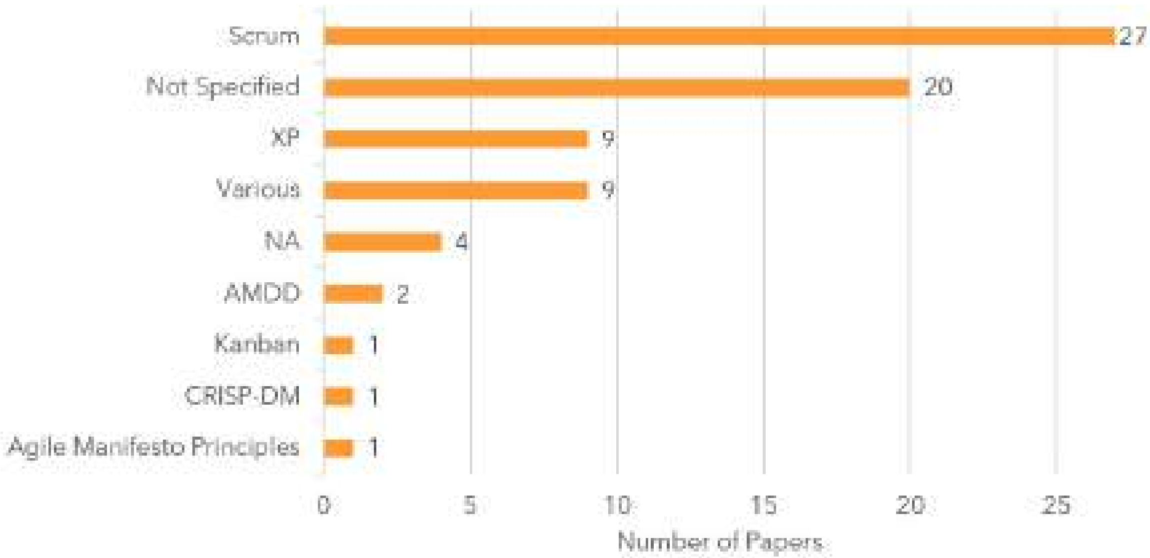


Figure 2.6 Distribution Agile Methods

Although majority of the researchers have limited the model integration to two methods, the Lean start-up methodology dominates the three method models (Fig. 2.7)

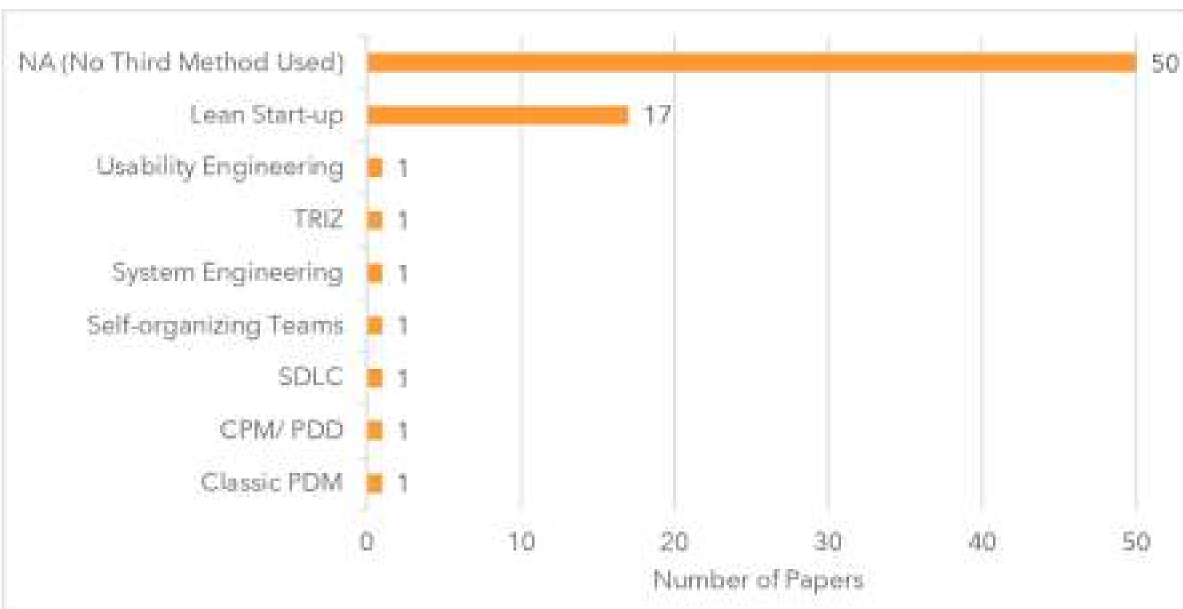


Figure 2.7 Tertiary Method usage in models

Most of the models use upfront design thinking methodology to collect and clarify user need at the start of the product development process. Once the user needs are clarified, the development team starts developing the product using agile methodologies (Figure 2.8).

18 models used the DT method throughout the development lifecycle. Only 1 model has used the DT model in the last phases of development cycle during testing and deployment.

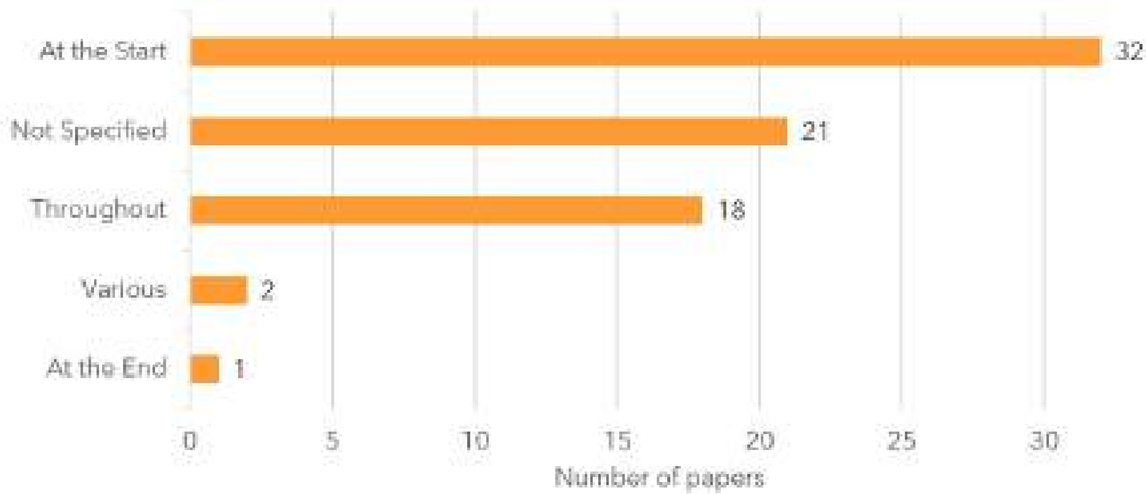


Figure 2.8 Method of model integration

The majority of the models, 57%, are conceptual in nature. 30% Models are verified with a practical application as shown in figure 2.9 below.

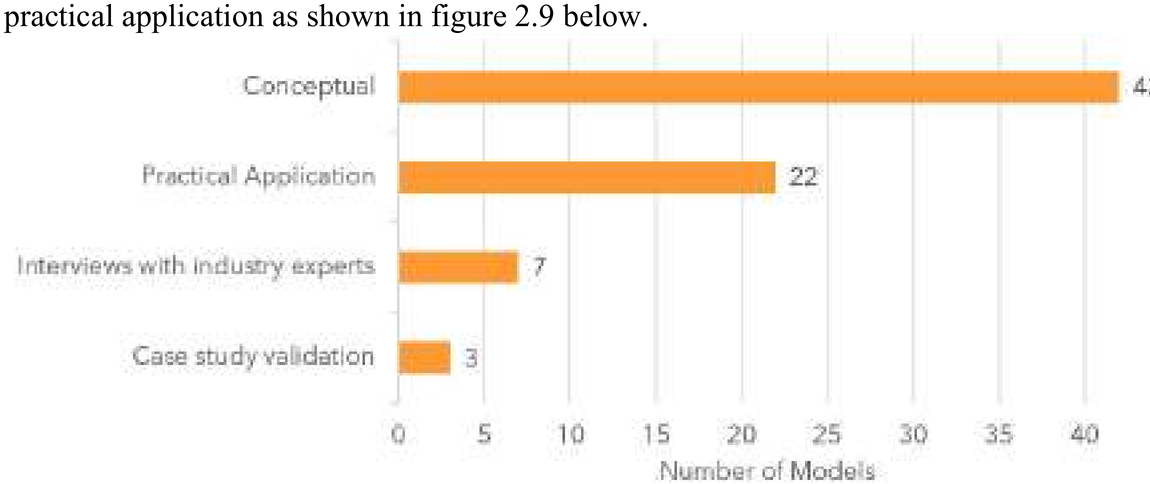


Figure 2.9 Model validation method

The literature research indicates various applications of the newly developed models as shown in table 2.9.

Table 2.9 Various model applications

#	Model Application	Frequency
1	AutoDesk Autocad Feature	1
2	Automation	1
3	Connected Health - e-Pharmacy	1
4	Cyber-physical systems	2
5	Data Analytics	2
6	Digital Innovation	1
7	Digitization in specific industry	1
8	Educational Game App	1
9	Educational Web App	2
10	ERP	1
11	Game development	1
12	General prodcut and processs design & development	3
13	Information Technology	3
14	Interactive real-time applications	1
15	Interactive software development	1
16	Internet Portal (e-commerce, mobile apps & SAAS tools)	1
17	Intuit's Quickbooks product line	1
18	Mobile App	2
19	Mobile game development in start-up	1
20	Mobile Multimedia Streaming App	1
21	Not Specified	2
22	Requirement Engineering	2
23	Safety Critical system	1
24	Sales portal and other applications	1
25	Service Design	1
26	Software development	28
27	Software Requirement Engineering	3
28	Software systems for people with complex disabilities	1
29	Web App	7

Finally, the literature research indicates various combinations of the DT and Agile models as shown in figure 2.10.

It is observed that the HPI with Scrum is the most used combination model followed by ISO & Scrum, and D.School & Scrum.

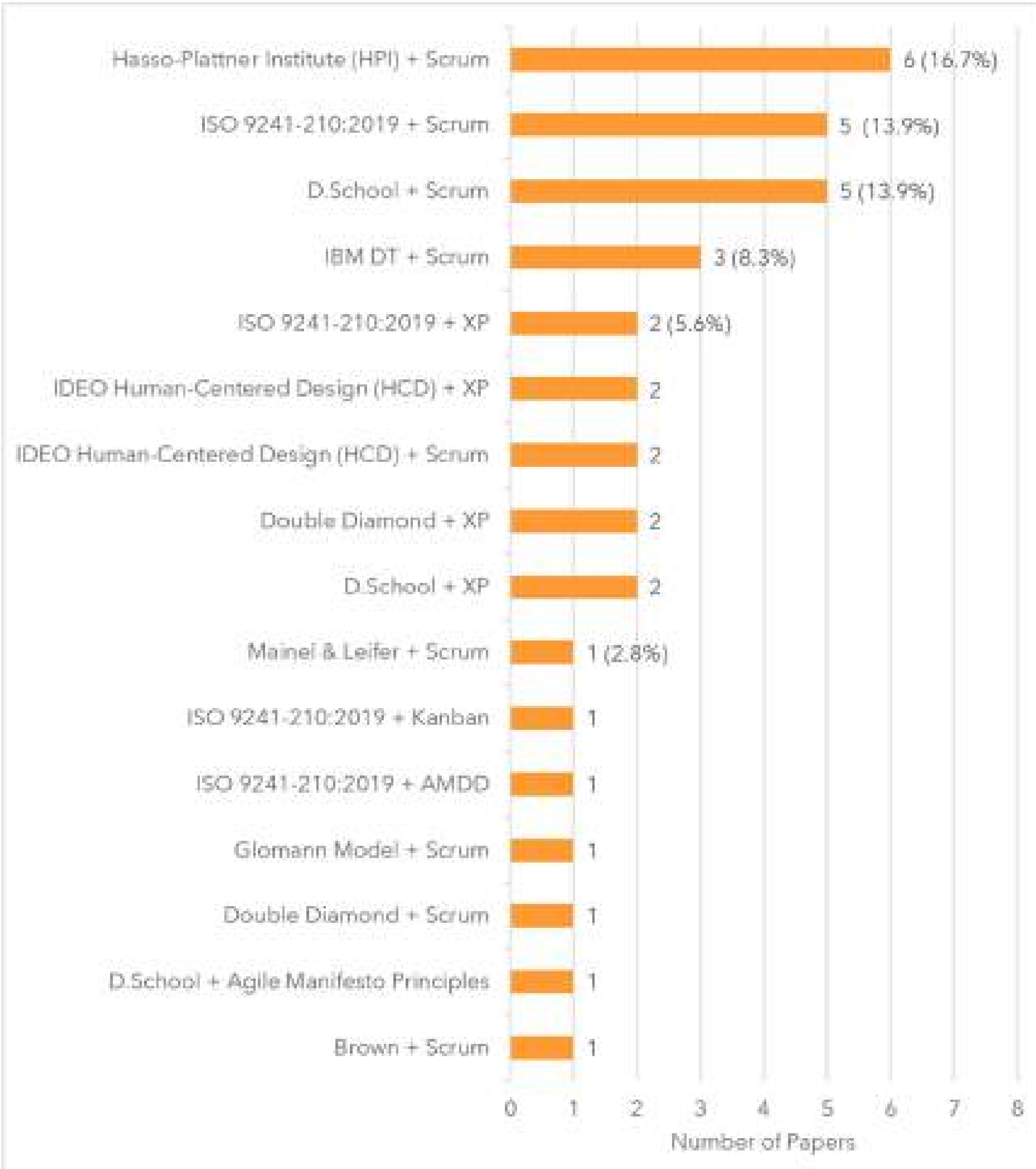


Figure 2.10 Model combination spread

2.9 Integrated model's overview – SLR Studies outcome

The literature review shows that prior studies are primarily focused on building models as a combination of Design thinking (DT), Agile and in few cases Lean start-up also.

Parizi et. al. (2022) conducted systematic mapping study to analyze the application of DT in software development. They provided criteria for DT model selection and observed that Upfront DT model used at the start of the development process for requirement engineering is the most used integration approach.

Pereira and Russo (2018) systematically studied various models combining Design thinking and Agile methodologies and presented the various combinations of both the approaches along with their application. They also analyze the type of integration adopted between the DT and agile such as sequential or throughout the software development lifecycle (SDLC).

Zorzetti et. al. (2021) conducted a systematic literature review on usage of User Centered Design and Lean Start-up in Agile software development. They used an approach similar to Systematic Mapping Study (Petersen et. al., 2008; Pereira et. al., 2018). They observed that there are very few models in which all three approaches are used. Those few models are created based on author's own methodology and a disconnect with industry is observed.

Almeyda et. al. (2021) have conducted a systematic review of literature available on application of integration of agile and user experience in requirement engineering. They observed the three best practices in requirement engineering such as Person/ user focus, low-fidelity prototype, and User stories.

Argumanis et. al. (2020) have conducted a systematic literature review to understand the challenges while deploying the combined Scrum and User Centered Design framework. They observed that, most of the challenges are associated with less focus on user experience and usability principles, inadequate resources assigned in Scum team to activities at the start of the program, and gap between the real user needs and client's pitch to the developers.

Martins et. al. (2019) conducted systematic Literature Review using Data triangulation method (Trivisios, 1987) for challenges in software requirement elicitation and how design thinking can be used to resolve those.

2.10 Model integration theory

There are three types of DT & Agile model integration observed such as:

1. Backward Integration: DT applied at the end during testing and deployment for capturing customer feedback as a part of Usability Engineering approach.

2. Forward Integration Approach: DT applied at the start during requirement engineering for better understanding of customer needs as a part of Requirement Engineering Approach.

3. Continuous integration approach: DT applied throughout the development cycle iteratively during all development phases.

Hehn et. al. (2019) suggested three approaches of DT integration in software development Requirement Engineering (RE) such as: Upfront, Infused and Continuous. Upfront DT is used before the RE to define product solution vision. The Infused DT is used within RE to create innovative ideas or clarify fuzzy conditions. And the Continuous DT is a combination of upfront and infused approach and used as a guide throughout the RE process. They suggested a combined model of Design thinking and Requirement engineering using a case study validation technique.

Lindberg et. al. (2011) observed that the upfront approach to use DT in IT projects is easy to deploy, however, there is a risk disruption in information flow from DT phase to subsequent agile development phase. On the other hand, continuous integration has advantages such as enhanced flow of knowledge in teams, leveraging diversity, and better understanding amongst team about the next developmental steps.

Gabriel et. al. (2021) created a hybrid model combining various agile methods and classical product development methods for intelligent system development. The hybrid product development model is an Agile Toolbox with fifty agile methods. The model can be tailored to a company specific program by analyzing the initial situation and matching & integrating the agile practices.

Parizi et. Al. (2020) recommended a tool for usage of DT technique in Requirement engineering in software development. They used a three-step model for DT tool recommendation such as "Meta design thinking session" (Problem definition and solution development), "Requirement elicitation" (User personas, journey maps, blueprint & low-fidelity prototypes) and "Early evaluation" (Interviews with experts)

Pereira et. al. (2021) conducted a study using focus groups and surveys to understand the impact of DT adoption in software development. It is observed that the DT results in better understanding of requirements, reduces uncertainty, increased collaboration and with understanding of business needs, the requirement log itself improves.

The above mentioned three approaches of model integration theory are explained in the subsequent sections 2.11, 2.12 and 2.13.

2.11 Backward Integration Models: Usability Engineering Approach

This theory explains the integration of DT at the end of the agile development lifecycle to seek feedback from customer or end user and subsequently improve the product.

Sohaib and Khan (2011) used DT at the end of the development process during Usability Engineering phase. Their research include application of grounded theory (Strauss & Corbin, 1997; Oktay, 2012) to actual software program team, which resulted in integration of Extreme Programming (XP) and Discount Usability (Nielsen, 1994; Butler, 1996) practices. Usability Engineering focuses on the use of systems by the customer combining aspects of human and machine interaction (Mayhew, 1999).

In product development lifecycle, they used various of aspect Discount usability (Nielsen, 1995) such as "Scenario usage with user stories", "Card sorting in release planning", "Heuristic evaluation in UAT (User Acceptance Testing)", and "Thinking a loud during production phase". The biggest advantage of this approach is that the developer gets a clear perspective from end user involvement.

2.12 Forward Integration Model: Requirement & Design Engineering Approach

The Forward Integration Model Theory proposes integration of DT tools and practices at the start of the development lifecycle during requirement and design engineering phases.

Wölbling et. al. (2012) recommended to use design thinking for user-centered software development emphasizing four key elements such as iterative nature, multi-functional team, creative space, and mindset of a designer. They also discussed limitations of DT application such as teams' reluctance and organizational resistance to change.

Grossman-Kahn (2012) developed a descriptive model in Nordstrom Innovation Lab involving DT, Lean and agile which is scalable and can embrace a newly emerging methodology. In this, the continuous delivery, Build-inspect-retrospect loop, and validated delivery framework are delivered by lean and agile. Whereas the roadmap to develop user-centered innovative solutions is provided by DT.

De Paula (2016) applied an improved version of Nordstrom model (Grossman-Kahn and Rosensweig, 2012) to an IT application (game) development program and found that, during prototyping phase, the SQA team should use the UI/ UX design closest to the definitive version. They also suggested that the DT approach should be used throughout the Scrum phase rather than at the start of the project.

Dobrigkeit and De Paula (2017) compared the "DT@Scrum" (Hager et. al., 2015) and Nordstrom model (Grossman-Kahn and Rosensweig, 2012) using BPMN (Business Process Modelling & Notation) method. Based on the best practices from both the models, a new methodology InnoDev is suggested, which is more versatile to apply to large organizations, SMEs, and small start-ups.

Dobrigkeit (2019a, 2020a) developed a model "InnoDev", which is an agile software development method combining DT, Lean startup, and Scrum. In this, Design thinking helps in understanding user needs and derive product solution options, and Lean startup helps in business sense and scaling strategies. InnoDev has three phases such as DT phase, Initial development phase and Final development phase depending on magnitude of ratio between usage of the three philosophies. They also developed a toolbox - "DT@IT Toolbox" (Dobrigkeit, 2020b) which can be used by software development teams in daily agile development work. It helps a novice team to enhance empathy, improve problem solving skills and better understanding of user needs.

Hildenbrand and Meyer (2012) used the design science approach to integrate Lean and design thinking for Enterprise Resource Planning Software development. Their model has three phases such as 360° Research, Synthesize (vision, story map and backlog), and development with Scrum. The transition from empathy to insights can be done by User Story map. Two key take-aways from their research are 'Stop to Think' and 'Expect waste in innovation.'

Adikari et. al. (2013) reframed Design thinking for Agile user experience design projects. The design thinking methodology was applied throughout the development process as a reframed context. It considers real industry holistic view to create reframed context which can be concurrently used for two frameworks such as user experience design and agile software development. At the end, the product is the combined result of both the frameworks.

Carroll and Richardson (2016) developed a process model for connected health sector for developing e-pharmacy system. The model has 3 stages: Problem in healthcare innovation (Design thinking), Needs for defining solution (Requirement development) and Solution integration and verification (Technical system development). The model integrates D.School design thinking model with a project management methodology - Waterfall or Agile. Considering human safety aspect in healthcare sector, design thinking helps in effective requirement engineering.

Higuchi et. al. (2017) have applied the combines model for highly unstructured game development project. It integrated the Scrum methodology with Brown's DT model "Inspiration – Ideation – Implementation" cycle via an overlapping connecting phase of product vision and product backlog. They observed the dual benefit of the model as DT can help in product differentiation and Requirement engineering at the initial phases (Concept and pre-production) when creativity is highly needed, and on the other hand Agile can help to enhance development efficiency during later phases such as production, deployment & post launch.

Darrin and Devereux (2017) have mapped the DT, Principles of Agile Manifesto and System Engineering steps, for engaging the users in whole product/ process development method which will help in reducing risk and increasing agility during the program execution.

Lucena et. Al. (2017) at IBM research lab developed a DT model which can be applied throughout the agile development process. The upfront approach of D.School DT model such as Understand (Empathy & Design), Explore (Ideate), Prototype and Evaluate (Test) was applied to understand user's requirements. Then 3 stage approach, Hills - Sponsor users - Playback, was iteratively applied throughout the scrum process. Hills express needs, Sponsor user is a real human to provide point view than theoretical personas, and Playbacks are checkpoints for feedback on business goals and hills.

Prasad et. al. (2018) applied DT to agile development practices by usage of principles such as "human-centered approach, visualizing, thinking-by-doing, diverging & converging synthesis and collaborative work". They interviewed industry experts, and the survey data was analyzed using Grounded theory (Oktay, 2012; Walker & Myrick, 2006) and found the key factors affecting customer satisfaction. Based on the factors the framework was suggested. They observed that early feedback from customers enhances the product quality and customer satisfaction with the product.

Przybilla et. al. (2018) developed a combined model using Scrum and DT for digitization program. They focused on the Human Centered Design (HCD) aspects such as knowledge management, Human/ user aspects, and challenging assumptions. The process was sequential and the key handover from DT to Scrums was done using three items: Core idea, User insights and process support/ modular development method.

Luedeke et. al. (2018) developed and tested a product and process development model based on Weber's theory of CPM/ PDD (Characteristics Properties Modeling / Property Driven Development) (Weber, 2005). The product characteristics are transferred from creative stages to technical agile product development stages using CPM plotting method and considering the stakeholders and product owners. Due to usage of fundamental product maturity transformation based on time of development in CPM, this model can be applied to various products such as physical systems and software engineering systems.

Hehn and Uebernickel (2018) suggested a 3-step process for use of DT in product development. The phases are Exploration, Alpha prototype, and Friendly user test & Market launch. They observed that the process provides a structured approach to requirement elicitation for a wicked problem. The method also improves team collaboration and usability by supporting the concurrent and upfront Requirement Engineering practices.

Sohaib et. al. (2019) have developed a software engineering project framework, "DT@XP", by integrating the design thinking method upfront with Extreme programming method. The framework constitutes four phases such as Exploration, Planning, Iteration to Release and Productionizing. The first phase includes design thinking steps and the rest of the three phases are related to agile development. Both the methodologies are integrated between DT user stories and Usability evaluation.

Brad et. al. (2019) have implemented TRIZ methodology for resolving barriers, conflicts and gaps while integrating the DT, Scrum and Competitive Engineering for software development. They developed a tool - Competitive agile-lean-design thinking (CALDET) for software development services. This tool removes the gaps in the integration of various methodologies and reduces fuzziness in the framework.

Blosch (2019) along with a team of technology innovation leaders and enterprise architects at Gartner, developed a model combining design thinking, Lean start-up and Agile for enhancing digital innovations. The shortcoming of this model is that it is sequential and has a tendency of waterfalling the product development process.

Ahmed et. al. (2019) created LDTM (Lean Design Thinking Methodology) for modern programs associated with Data mining and analytics application. They integrated the Design thinking for requirement engineering, Lean start-up principles for evolving model solution, and CRISP-DM (Chapman, 1999; Schröer, 2021) for developing algorithm of the model. They observed that, as the data evolves during the development cycle, the process becomes iterative and there is generally not just one way to be used in each phase, rather based on the development status, different approaches can be used in each phase.

Alhazmi and Huang (2020) integrated D.School design thinking model with Scrum agile framework for effective requirement engineering. The framework helps in requirement engineering at various levels such as requirement elicitation, analysis, documentation, validation, and management. The paper presents a detailed analysis of challenges faced in requirement engineering and how the new framework supports dealing with challenges. The model helps in enhancing the team's creativity and productivity, requirement definition clarity and product quality. Signoretti et. al. (2020) have tested Alhazmi's integrated model with two agile teams, and they observed that the new methodology implementation success is based on project team's level of engagement and mindset.

Magare et. al. (2021) developed an integrated architectural framework called "Intelli-A" which maps design thinking process with agile framework. The framework is multilayer with the inner 2 meta-layers associated with design thinking and three canvases such as Mind mapping, empathy map and AEIOU (Activities, Environments, Interactions, Objects, and Users) (Hanington & Martin, 2019). The middle two layers are of agile methodology across product development canvas and ideation canvas. In this, the product prototype is built, tested and user feedback is received. The outermost layer indicates all the element's categorization into various stages of SDLC (Software Development Lifecycle).

Marion et. al. (2021) suggested a conceptual upfront type of integrated model of DT and lean start-up. They created a model in the form of double diamonds dividing into problem space and solution space. The Problem space includes design thinking steps with focus on problem articulation and selection. The solution space includes Lean start-up steps with focus on concept generation and agile methods with tasks such as concept selection, development test etc. The model is an effective way to integrate processes such as customer need identification, ideation, and development with user in focus.

Moralles (2020) developed a model combining the various principles and practices from Agile, User Centered Design (UCD) and Lean Start-up. Zorzetti et. al. (2020, 2022) also presented a similarly combined model for agile software development using methodologies such as XP, Double Diamond Design thinking method and Lean start-up philosophy backed by User Centered Design (UCD) approach. There are three phases: Discovery, Framing and Iteration. In each phase, lean BML (Build - Measure - Learn) cycle is used. In Discovery phase, the team get a deeper understanding of the user requirements mainly through interview and early feedback. In the Framing phase, different solution options are explored, the solution is chosen, and MVP (Minimum Viable Product) is built. In the Iteration phase, agile development work using XP is done. Signoretti et. al. (2020) have tested this model with two agile teams, and they observed that the new methodology implementation is based on two critical factors such as team engagement and problem solution-oriented mindset.

2.13 Continuous Integration Model: Full Development Cycle Approach

The Continuous Integration approach consists of usage of Design Thinking throughout the agile development cycle.

Glomann (2018) developed a continuous integration model – HCAW Model (Human-Centered Agile Workflow). The DT phases are fully integrated throughout the development lifecycle at all the stages. Each cycle 'n' consist of multiple sub-phases such as “Cycle Conception”, “Sprint 1 and 2 Conception” and “Sprint 'n' Development”. In the Cycle 'n' phase, all the DT phases such as Research, Ideation, Prototyping and Evaluation are carried out. Then the Cycle Plan is transferred to Sprint conception, where conception phases such as Ideation, Prototyping and Evaluation are carried out resulting Specifications (Story Grooming). And at the end of the sprint, the Implementation phases such as Business, Development and Quality Assurance are carried out. The major benefit of this process is customer-centered RE resulting in high quality business value.

Ximenes (2015) developed a model, Converge, by combining Agile, DT and lean start-up along with the empirical observation from a software development lab. They have also integrated best practices from Extreme Programming (XP) such as Collective code ownership and pair programming file executing feature-by-feature implementation.

Gurusamy et. al. (2016) proposed a framework to support the ongoing digital transformation by combining DT with agile methodologies in a continuous manner. They recommended a 3-phase process – Requirement, Design and Evaluation throughout the agile cycle. The Requirement phase include understand, observe, Point of View (POV) and ideate. The Design and Evaluation phases considers prototyping and testing, respectively. The authors recommended focusing on team coordination at various levels in future.

Nedeltcheva and Shoikova (2017) have applied the IBM Design Thinking process to AutoDesk use case of a feature development in AutoCAD. With the application of IBM

DT, they observed better ROI, Collaboration, elimination of duplication and product improvement.

Lukasik et. al. (2018) at IBM institute, suggested an integrated model by deploying "Hybrid Sprint", which includes design thinking task at the start of each sprint. Continuous integration results in enhanced product quality, better alignment with stakeholders and on-time customer feedback.

In this way, it is observed that majority of the research work is conducted on Forward integration type models followed by continuous integration models.

2.14 Literature Review Summary

During this research work, a novel framework was developed for conducting literature review - “Systematic Integrated Mapping Method” (SIMM). The method is a combination of Systematic Mapping Studies of literature, Benchmarking, and Integration of the analyzed data. A search string is defined, and a search is conducted in digital libraries resulting in 2293 research papers. The literature search period is for 23+ years from year 2000.

The first level screening includes various exclusion criteria such non-peer reviewed, non-English, and duplicate papers. The second and third screening criteria are reading title, abstract & keyword and reading full paper, respectively. Based on three level screening, finally 3.2% papers i.e., 74 papers, which provides models and philosophies, were selected for detailed literature review.

Three model integration methods were suggested such as Forward, Backward, and Continuous integration. In Forward Integration Approach, DT applied at the start during requirement engineering for better understanding of customer needs as a part of

Requirement Engineering Approach. In Backward Integration, DT is applied at the end during testing and deployment for capturing customer feedback as a part of Usability Engineering phase. In Continuous integration approach, DT is applied throughout the development cycle iteratively during all development phases.

This research work answers the first two research questions. A detailed quantitative analysis of the literature is provided. Then the whole literature is studied, analyzed, synthesized, and categorized into the above mentioned three model frameworks.

2.15 Conclusion

Based on the detailed literature study of models & philosophies developed by various authors, it is observed that there is not sufficient work done to create models for developing ultra-complex cyber-physical systems.

Major gaps are observed in integration methodology, the application, Model combinations and tertiary models.

Following is the summary of research gaps from literature review:

► Most of the authors have used the Forward approach of modelling specifically for requirement engineering only. Whereas a full-scale model with the application of DT at all the stages of product development lifecycle is needed. However, a very limited application of full-scale model is observed.

► The tertiary model is not used in most of the papers (67.6%) and only lean start-up philosophy is observed in a few papers (23%).

► There is very limited work done on model development for new-generation ultra complex technology development such embedded cyber-physical system, AI & ML algorithms development and Generative AI programs etc.

- ► Majority of the work use Scrum (65.8%) methodology and 55% work use the D.School, ISO or HPI as design thinking methodology. So, it is observed that very few novel methodologies are developed.
- ► The most popular model combinations are HPI with Scrum (16.7%), ISO & Scrum (13.9%) and D.School & Scrum (13.9 %). There is no study conducted which analyses and presents the pros and cons of the various model combinations.

As a next step, the knowledge gained in this literature study is leveraged and list of pitfalls is created associated with software development project in context of various integrated models used for development work.

CHAPTER III:

METHODOLOGY

3.1 Overview of the Research Methodology

This chapter discusses in detail about the research methodology adopted. It talks about the research purpose & questions, research design, process of sampling, data collection & analysis.

The step-by-step methodology used for conducting this research is depicted in figure 3.1. The research started with framing the research question - the big question - what are we trying to achieve during the course of this research? For the specified question, the next step is to conduct a detailed review of the work already conducted by the other researcher from the year 2000. We developed a novel methodology for conducting the literature review.

Next, the project performance data was collected from one of the Fortune 500 company & analyzed to understand the problem areas/ functions for project failure, and key reasons for project delays. Then various product development methodologies as practiced in industry are plotted as block diagrams and their characteristics and limitations are noted. Based on the analysis of the data, insights from the detailed literature review and study of various product development methodologies, a list of pitfalls on programs was created.

Variables were defined as dependent (Y), sub-dependent (y_i) and independent (x_i) for each pitfall. A survey questionnaire was created based on pitfalls and variables. The data from the survey was statistically analyzed and a relationship was formed between various factor and project success. Based on the complete study, factors relationship and statistical analysis, a conceptual model was proposed along with recommendations for deployment.

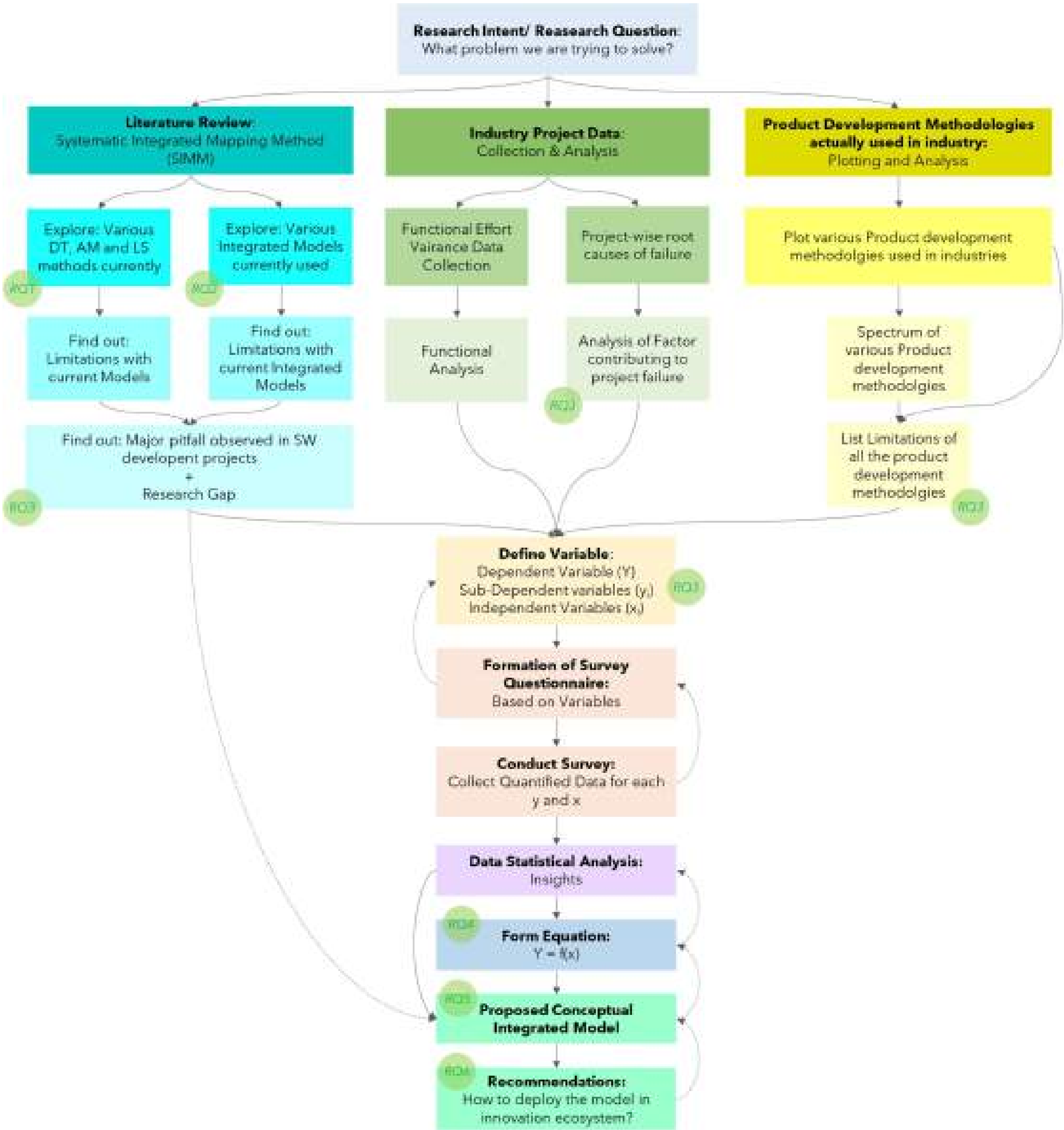


Figure 3.1 Overview of Research Methodology

3.2 Operationalization of Theoretical Constructs

In this research the theoretical construct are abstract concepts such as agile, lean, design thinking etc. To operation these theoretical construct means connecting the theory to the real-world data by defining measurable variables (Carmines and Zeller, 1979). The variable we are looking at are on-time delivery, cost-adherence, quality, customer satisfaction and reduced scope creep.

Agile focuses on iterative product development (Beck et al., 2001), Design thinking emphasizes on empathy and user-centric approach to requirement management (Brown and Wyatt, 2010) and Lean philosophy focuses on early testing of business hypotheses by deploying BML (Build-Measure-Learn) loop (Ries, 2011). The dependent variables are operationalized using the metrics used in software process assessment (Unterkalmsteiner et al., 2023). Based on the three studies such as literature review, industry data analysis and study of product development methodologies, various independent variable are defined. To evaluate impact of various predictors, we used survey method with improved Linkert scale (Linkert, 1930) which provides 6-point ordinal measurement with more sensitivity as there is no midpoint (Joshi et al., 2015; Boone, 2012). The quantification and analysis of the data helped in quantifying the relationship and operationalize the construct.

3.3 Research Purpose and Questions

The purpose of this research is to study and understand various factors which governs the software development projects, build predictive model for predicting the project success, and build a software development model combining best practices from various philosophies.

Following research questions were formed:

- RQ1: What contemporary Design thinking, Lean thinking and Agile methodologies are practiced in software development Industries?
- RQ2: What are currently available 'integrated models' developed by researchers?
- RQ3: What are pitfalls and corresponding governing variables in current ultra-complex product development programs?
- RQ4: What is the relationship between various independent variables and program success? How can the project success be precited based input parameters?
- RQ5: How various philosophies can be integrated to build up a model to eliminate the common pitfalls?
- RQ6: How can the conceptual integrated model be deployed in innovation ecosystem?

The figure 3.1 helps to visualize the instances of each research question on research process flow.

3.4 Research Design

Descriptive research was conducted to understand how software development projects fail and to find the relationship between the various governing factors and project success. Data was collected from live projects as well as surveys to quantify the variables. As shown in figure 3.1, the Survey questionnaire was created based on the variables (dependent and independent) identified. To define these variables, a three-way approach was used – Industry project data analysis, Pitfalls observed in current process of project based on literature review, and plotting & analysis of various project management processes observed in industry.

3.5 Population and Sample

For our survey, the participants are predominantly program managers. As per Project Management Institute (PMI, 2023), there are currently approximately 39.6 million project management professionals. Other participants include similar roles such as System engineers, Technical architects, Software project managers etc. If population size is unknown or greater than 10,000, then it can be classified as infinite population (Cochran 1977; Israel, 1992). So, the population for this survey is considered to be infinitely large.

For sample size calculation, we deployed various methods based on the type of statistical analysis we will perform on the data.

Method 1: As per Green's Rule (Green, 2015) –

1. The sample size for the Multiple Linear Regression analysis for testing the overall model (i.e., R^2 significance) is:

$$N \geq 50 + 8m$$

Where,

N is Sample Size

m is Number of Independent Variables = 15

Hence, $N \geq 50 + 8 \times 15$

$N \geq 170$

2. The sample size for testing individual predictors (i.e., each β coefficient) is:

$$N \geq 104 + m$$

Hence, $N \geq 104 + 15$

$N \geq 119$

Method 2: As per Cochran (1977), the formula for sample size calculation for infinitely large population is:

$$N = \frac{Z^2 . p(1-p)}{e^2}$$

Where,

N is Sample Size

Z is Z-score (1.96 for 95% confidence)

p is estimated proportion of the population (Example: 0.5 for 50% of population)

e is margin of error (0.05 for ±5%. This value is robust and widely used in industry as it provides good balance between feasibility and accuracy)

We have tried for three different values of proportion of population and calculated the sample size as follows:

Table 3.1 Sample Size Calculation using Cochran Formula

Scenario	Confidence Level	Z-Score	Population Proportion (p)	Margin of Error (e)	Sample Size Calculation (N)	Sample Size (N)
1	95%	1.96	10%	0.05	$\frac{1.96^2 \times 0.1(1-0.1)}{0.05^2}$	138
2	95%	1.96	30%	0.05	$\frac{1.96^2 \times 0.3(1-0.3)}{0.05^2}$	323
3	95%	1.96	50%	0.05	$\frac{1.96^2 \times 0.5(1-0.5)}{0.05^2}$	384

The average of sample size of the above three scenarios is:

$$\text{Sample Size} = \frac{138 + 323 + 384}{3} = 281.66 \approx 282$$

Method 3: Using G*Power Tool (Version 3.1.9.7)

We used following parameter values as input to the software:

Effect Size f^2 = 0.15 (Based on Cohen's guideline for moderate relationship between predictors and response)

A error probability = 0.05 Significance Level (Considering 5% risk of Type I error)

Power (1-β error probability) = 0.80 (Considering 20% chance of Type II error)

Number of Predictors = 15 Independent variable.

After entering these values, the Sample Size is calculated = 139 (Refer screenshot of G*Power tool below in figure 3.2)

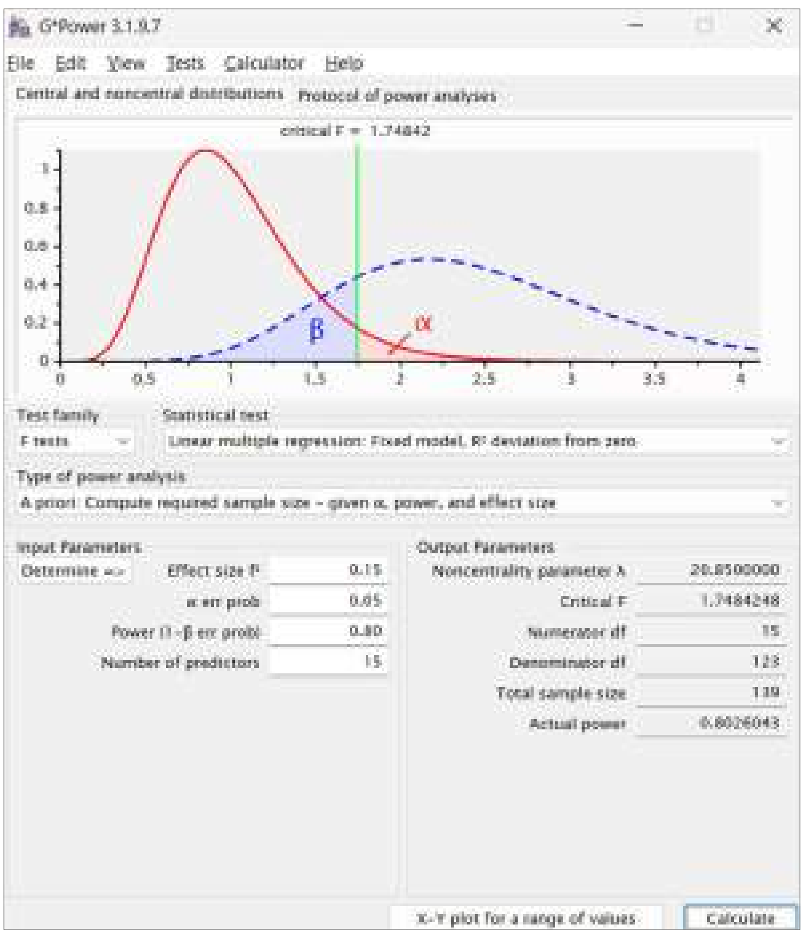


*Figure 3.2 G*Power Tool – Sample Size Calculation*

In summary, the three methods indicates different sample sizes as shown in table 3.2 below.

Table 3.2 Summary of Sample Size Calculation Using Various Methods

Method #	Method Name	Calculated Sample Size
1	Green's Rule - for testing the overall model (i.e., R^2 significance)	170
	Green's Rule - for testing individual predictors (i.e., each β coefficient)	119
2	Cochran's Formula	282
3	G*Power Tool	139

We selected the largest sample size of 282 to have better accuracy and feasibility of sample with respect to the population.

So, Sample Size for our survey is:

$N \geq 282$

3.6 Participant Selection

The survey was conducted on professional from industry, who predominantly work on software project management, software development, system design, engineering management, cyber-physical system projects etc. These participants were selected considering the need for getting better inputs and making the survey results more relevant for the research work. The professional social networking websites such as LinkedIn and author’s industry network were leveraged to source participants.

3.7 Instrumentation

The major instrument to collect the quantitative data was online survey. The survey was designed using improved Linkert scale to quantify the data on each question. Objectivity was maintained in questions to collect the data in accurate and reliable manner. This will make sure to minimize bias and errors in data collection. This will also help in arriving at an actionable insights post data analysis. One open ended question was added to collect any additional qualitative inputs (if any) from participant which were not included in the objective questions.

On the other hand, the project data was collected from real projects from an organization. This helped in gaining a direct insight on project failure functional analysis and exact reasons for project delays. Anonymity of the organization and projects was maintained considering the sensitive nature of project failure data.

3.8 Data Collection Procedures

The data was collected using an online survey considering the empirical nature of research. The survey questionnaire was sent to the participants on professional forums such as LinkedIn, personal email requesting feedback on survey and on alumni/ industry groups (Sue & Ritter, 2012). The email ID feed was optional to maintain optional anonymity of participants and encourage response from participants. We provided, at the start, the context and purpose of research, assured confidentiality an anonymity, author's contact information for any questions, and voluntary nature of the survey (Babbie, 2020). We kept the survey open for at least 2 weeks, sent gentle reminders to the participants periodically and monitored the response trend (Dillman, 2014). The mode of conducting survey was online as it is very cheap, efficient, fast, scalable & geographically accessible to large number of participants (Wright, 2005; Evans and Mathur, 2005) to collect the data.

3.9 Data Analysis

The data analysis started with cleaning and coding of the data. After checking normality of the data, the Exploratory Data Analysis (EDA) was performed to understand basic statistical parameters such as Mean, Standard deviation, Range, Median etc., descriptive statistics and visualization plots. Initial correlation analysis and Regression linear modelling was performed on the clen data. To test hypothesis testing, ANOVA, t-test, Linear regression coefficient methods were deployed. At the end the model Diagnostics & Validation using Multi collinearity, QQ plot etc. was performed. Finally, the analysis results were reported by R^2, p-values, Hypothesis decision.

Apart from Multiple Linear Regression (MLR) model, we developed various models such as Robust linear Model Regression, LASSO Regression Model, ANOVA, Structural Equation Modeling (SEM), Random Forest Model, XGBoost Model, Support Vector Regressor (SVR), Multi-layer Perceptron (MLP) Regressor, SHAP (SHapley Additive exPlanations) Model, LIME (Local Interpretable Model-agnostic Explanations), Model Decision Tree (Supervised Learning ML Model). Based on analysis from all these models, we listed various critical independent variables (x_i) which governs project success (Y)

3.9 Research Design Limitations

We have collected the data using online structured survey which is very efficient and scalable, however, it relies on self-reported information from the participants. This limitation of bias on the part of subjective interpretation (Podsakoff, 2003) is arrested by an improved Linkert scale which maintains consistency and reduces variability in the data. The cross-sectional nature of the survey results into collection of data at one point in time over longitudinal surveys over very long period of time (Rindfleisch, 2008). However, this research design helps to capture the voice from current practitioners on latest industry practices. Also, on participants selection, we have selected industry professional who have relevant experience in the software development & project management field. This may lack the generalizability of findings to broader industry (Etikan, 2016), but at the same time, it serves the purpose of current research objective with inputs from most highly relevant participants.

3.9 Conclusion of Research Methodology

In this chapter, we elaborated the research purpose & questions followed by the research design. We also calculated the sample size using various appropriate methods and discussed about the process of sampling, participant selection, and data collection. We provided high level overview of the techniques used for data analysis along with expected final output reporting. Based on the insights from data analysis and modelling, we will build a software development project framework. At the end, we discussed about the limitation of research design.

CHAPTER IV:

RESULTS

4.1 Results from study of product development methods in industry

We have studied product (software and cyber-physical) development methodologies in various industries. We studied the project management and product development methodology in 10 different product companies and based on observation, plotted the methodologies in 4 types as below:

Classical Waterfall Methodology

A waterfall approach fitted for high-tech product development, which is commonly observed in industries is depicted in figure 4.1. This approach is generally used for development of complex electro-mechanical, mechatronics, automobile, traditional simple software products.

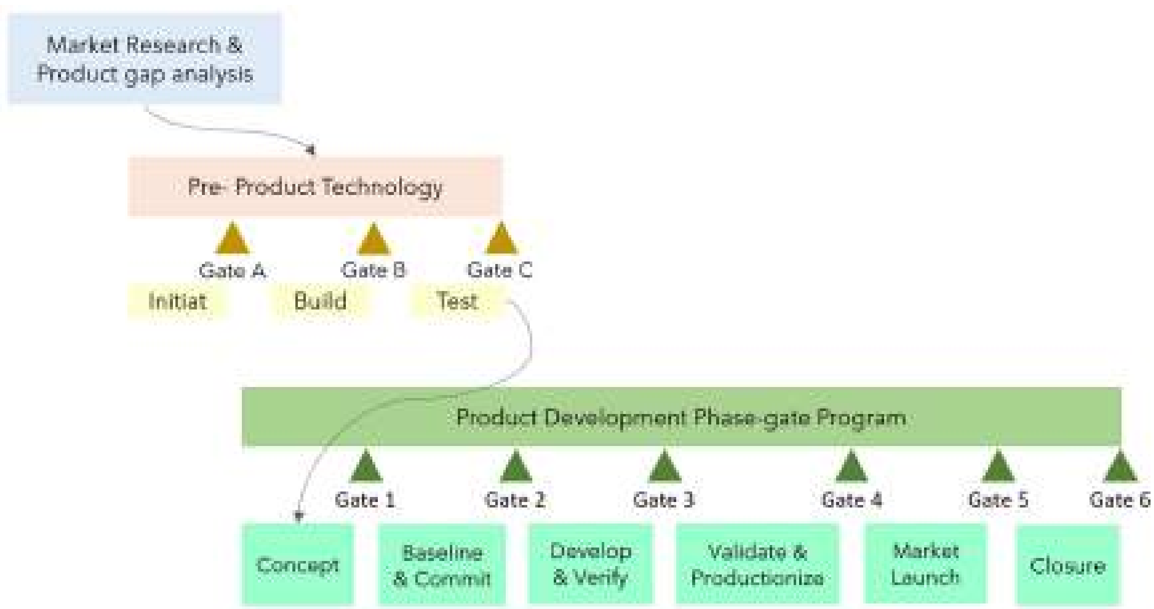


Figure 4.1 Waterfall Approach to Product Development

Hybrid Methodology

A Hybrid approach, observed in industries across globe, for developing cyber-physical products is shown in figure 4.2. This approach is generally used by companies for developing cyber-physical products which combines technologies such as software, firmware, electronics, mechanical, electrical etc.

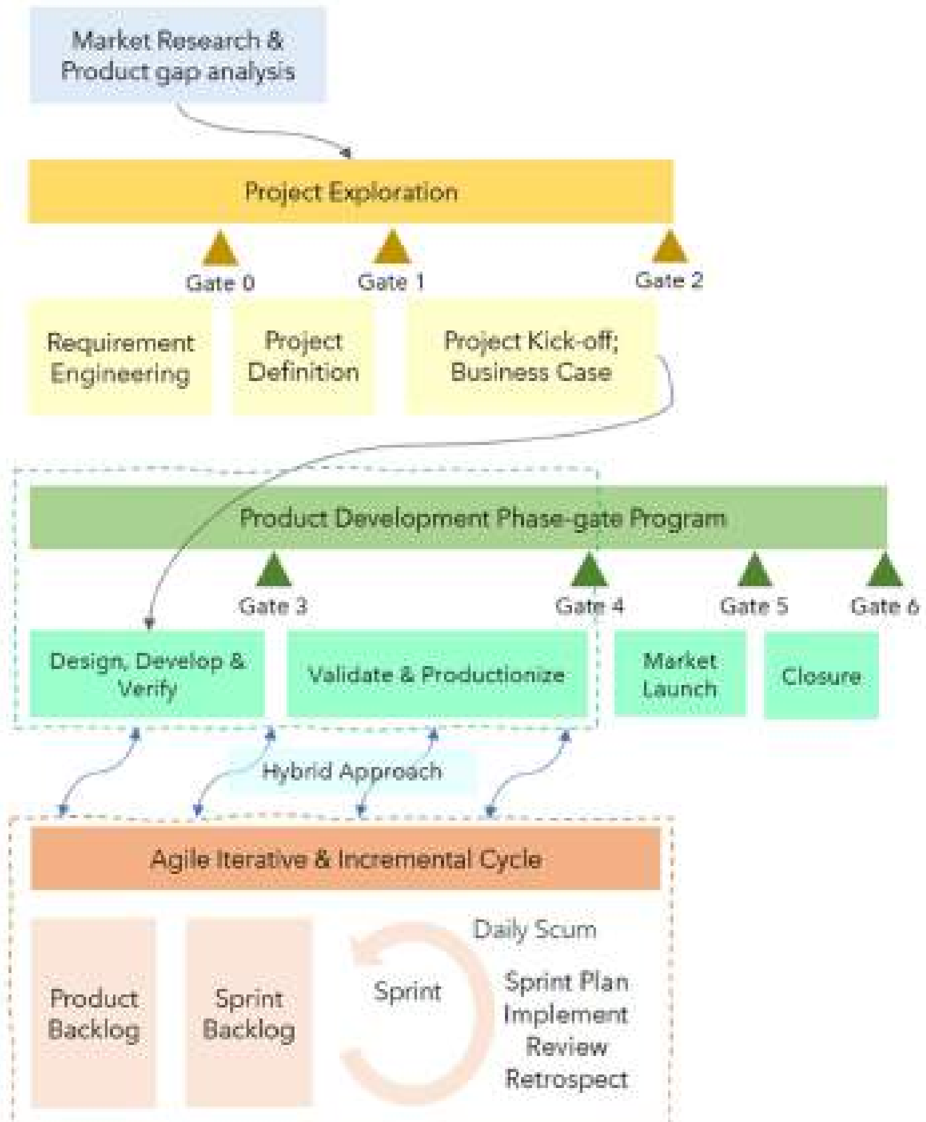


Figure 4.2 A Hybrid Approach for Cyber-physical product development

Agile – Scrum Methodology

It is observed that, pure software companies who build small web app, mobile app etc., where the resource are limited to one scrum team use Scrum methodology, as shown figure 4.3. This methodology was developed by Schwaber and Sutherland in 2001signing agile manifesto (Schwaber and Sutherland, 2020).

Figure 4.3 Pure Agile Methodology – SCRUM (Schwaber and Sutherland, 2011)

Agile – Scaled Agile Framework (SAFe)

There are companies who are in business of complex software development such as AI/ ML software where the resources are spread across globe with multiple scrum team working on multiple pieces of software deliverables at the same time. SAFe model is used by such companies. Dean Leffingwell developed the Scaled Agile Framework in 2011 and since then, it is used by various high-tech companies (Knaster & Leffingwell, 2020; Wilmshurst & Quick, 2023). SAFe framework is show in figure 4.4 below.

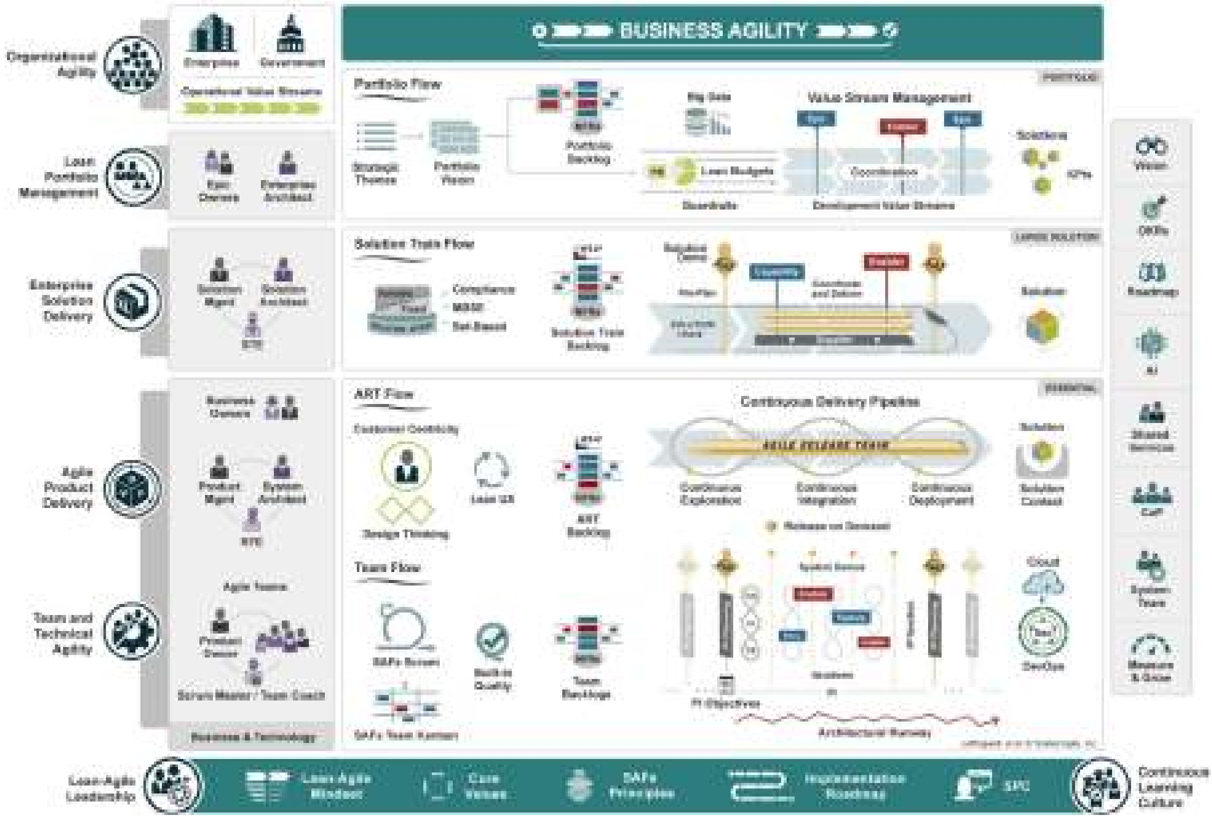


Figure 4.4 Scaled Agile Framework (SAFe) (Dean Leffingwell, Scaled Agile Inc.)

The methodologies are plotted on a spectrum showing the methodology a typical application as shown in figure 4.5.

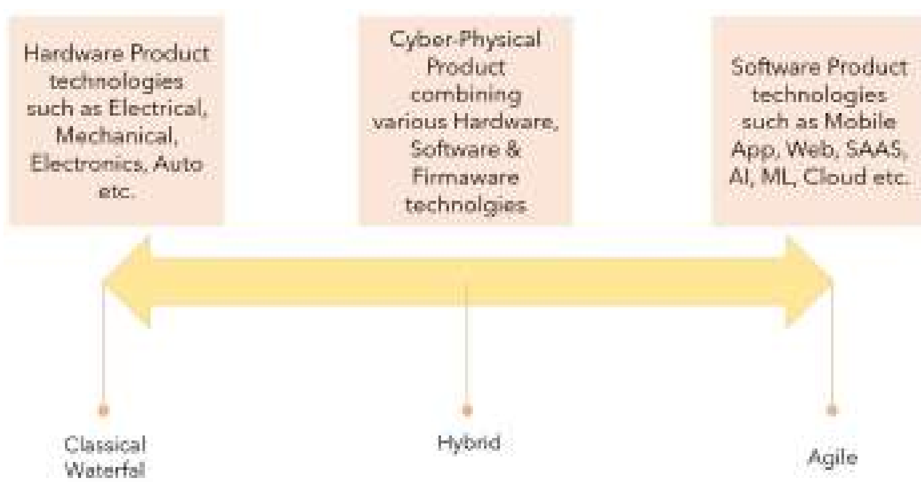


Figure 4.5 Spectrum of Product development methodologies

4.2 Results from Analysis of Agile Software Development (ASD) Process

From the detailed systematic literature review, we analyzed the shortcoming and limitations for various Agile Software Development processes. Various commonly observed pitfalls in using pure agile methodologies for software development are mentioned below:

► **Lack of deep understanding of customer problems** due inability to empathize with customer for their thought process and needs (Aulet, 2013). The team may develop a product which may not be able to solve a customer's problem (Tobak, 2014). This may impact profitability in future and high payback period (PB), Low Return on Investment (ROI), Low Internal Rate of Return (IRR), and low or negative Net Present Value (NPV)

► **Solving a wrong problem.** This may result in developing a product which customers do not need (Griffith, 2014, Ries, 2011; Blank, 2013, Christensen, 1997). This may result in loss of valuable time of teams, cost & efforts ultimately resulting in losing competitive advantage (Cooper, 2019) and even complete business failure.

► Sometime in product companies who sell software as a product to mass market, the **requirements do not come from real user**. Rather the requirements are derived based on needs of a typical to extreme user (Ximenes, 2015; Jarke et al., 2011; Sawhney et al., 2005). This adds challenged to the Requirement Engineering phase of the project.

► Even though the team build the product, many companies **lack on the right GTM (Go-To-Market) Strategy** as a result of insufficient competition study (Deeb, 2013; Keller and Kotler, 2009).

► **Insufficient scope definition** by project team, which results in frequent scope change and excessive product backlog refinement in future ultimately resulting in scope creep (PMI, 2017; Morris, 2013). Even for pure agile projects, insufficient scope definition and product vision results into frequent reprioritization of stories and backlog refinement (Cohn, 2004) resulting into negative impact on project success (Serrador and Pinto, 2015)

► **Missing the human aspect in the design.** Due to rush in breaking product requirements into user stories and initiating sprint planning, agile teams may miss to consider the aspect of human-machine interaction (Lukasik and Saylor, 2018; Ferreira, 2011). So, the product may not be user-friendly resulting in loss of business sustenance.

► Incorrect Resource Planning and project estimation. **Due to lack of awareness of the end-to-end product obligations,** the agile teams incorrectly calculate efforts resulting in inaccurate allotment of resources and budget, and lack of forecasts (Boehm, 2011; Molokken and Jorgensen, 2003)

► **Lack of sufficient documentation.** Due to the agility of the process, there is overall less emphasis on the formal project documentation (Ramesh et al., 2010; Inayat et al., 2015; Fitriani et. al., 2016). For example – Due to continuous feedback from Software Quality Assurance (SQA) team, the product owner may not follow the formal change management process.

► **Fragmented non-integrated outcome.** The various agile teams create fragmented outcomes at each sprint due to lack of clarity on exact product solution needed to satisfy human needs (Boehm and Turner, 2003). The fragmentation can also increase the risk of communication gaps between teams (Pikkarainen et al., 2008).

► **Lack of collaboration between various agile teams**. Due to varied understanding of product requirement, especially in case of global virtual teams now a days after COVID-19 pandemic, the global scrum teams lack effective collaboration, even after deploying Scrum of Scrums (SoS) ceremonies (Hossain et al., 2011; Stray et al., 2018).

► Ineffective monitoring of project health. Due to lack of vision about the exact result of project, the team may **not be able to properly define project key performance**

indicators (KPIs), resulting in absence of performance measurement, vision and project getting off-track (Kerzner, 2017; Crawford and Bryce, 2003)

► **Slipshod program governance**. Due to the characteristic of Self-managed Teams (SMT), there is a perception that the scrum teams do not need a strict governance (Goulstone, N.D.; PMI, 2017; Müller, 2009), which may result into loosing direction, elevated risk and low strategic fit (Dikert et al., 2016).

► **Non-suitability of common agile model to new generation technologies** such as AI, ML, Generative AI etc. The outcome of these projects is an algorithm based on the insights derived from data rather than programmatic codes (Walch, 2020; Singla et al., 2019). Also, the more traditional descriptive methods such PMBOK (PMI, 2022) miss the guidance on ethical complexity, data science lifecycle and iterative nature of AI-based Algorithm development projects (Burdakov et al. 2025).

► The combined models with Design Thinking and agile practices do not focus on tracking **growth and increasing scalability post product launch.** (Dobrigkeit & De Paula, 2017; Vilkki 2010; Grossman-Kahn and Rosensweig 2012).

►A typical agile methodology **lacks focus on design** (Dybå and Dingsøyr, 2008; Dobrigkeit et. al, 2019a) which may result into unsatisfied customer and rework in future. Typical design focus missed is on UX design-based specialist positions and Human-centered design (Curcio et al., 2019) and creative approach in story boarding and Requirement Engineering (Aldave et al., 2019).

►Agile teams sometime are so focused on incremental value delivery that, they cannot track the overall **impact of iterative nature on customer experience** (Molamphy, 2025; Lukasik & Saylor, 2018).

►Filippov et al. (2012) found that, **lack of strategic alignment of project portfolio** results into low project portfolio performance and steering. The alignment between portfolio and organizational strategic objective is significant in maximizing the projects values (Iamratanakul et al., 2008).

► **Lack of formal issue/ defect management process** result into lower software performance and process efficiency (Pai et al., 2021). This issue can be arrested by deploying a defect identification and recurrence prevention process in Verification and Validation and system engineering (Silva et al, 2017).

4.3 Results from Analysis of Integrated Design Thinking Models

► Even though the integrated models claim to be flexible, many of them appear to be **linear, sequential and are applied very rigidly** (Kettunen and Laanti, 2008; Brown, 2009).

► The design thinking part at the start of integrated models dominates the Requirement Engineering phase, hence it often focuses on individual problems rather than system as a whole resulting into **lack of system thinking**. (Kolko, 2018; Meadows, 2008; Buchanan, 1992; Johansson-Sköldberg et al., 2013).

► The emphasis is mainly on early-stage ideation, prototype, solution and feedback rather than end transition activities. There is an **insufficient Emphasis on Implementation – Execution and scaling** (Kalenda et al., 2018)

► Reflection and iteration based on real outcomes are not always implemented resulting into **limited Feedback Loops**. Beckman & Barry (2007) focuses in need of close looping reflection and integrating the outcome back to the design. Many firms fail to close loop process at may milestones such as experimentation, Software delivery and real outcome for customer (Brown and Katz, 2011). This problem related to feedback looping is enhanced when organizations scale (Paasivaara et al., 2018)

► In design thinking integrated models, the **cultural, ethical, and ecological aspects are often overlooked** (Frauenberger et al., 2017). Overreliance on persona definition, very slim testing specially affects the contextual and ethical considerations in a project (Goodman et al., 2012)

► While the upfront approach to use Design Thinking to understand the customer requirement in IT projects is easy to implement, but there may be **barriers in information flow from DT phase to subsequent agile phase** (Lindberg et. al., 2011).

► For Integrated Design Thinking implementation, there is generally a project **teams' reluctance and organizational resistance** to change observed (Wölbling et. al., 2012), which calls for a formal process to remove the impediment.

4.4 Industry Data - Effort Analysis and Insights

We have collected the project performance data from a high-tech product company, which manufactures the complex, cyber-physical products involving various technologies such as software, firmware, electrical, electronics, mechanical hardware etc. The data we have collected is functional effort variance for 11 projects.

Table 4.1 indicates the function wise effort variance data for each project. If the variance is positive (Overrun), it means, the team has spent more efforts than the baseline plan. Negative values (Underrun) indicate a better situation with less efforts spent on project with respect to the plan.

For this data, basic statistical parameters are calculated as below:

Inter-Quartile Range (IQR) calculation:

QTL_1 = The value at which 25% of the data lies below it.

QTL_3 = The value at which 75% of the data lies below it.

$IQR = QTL_3 - QTL_1$

Table 4.1 Function wise effort variance data for project

Project	Electrical/ Electronics	Mechanical	Firmware	Software	SQA/ SW Testing	Other (PM, EM)	Tech Comm	Total
A	16.1%	46.6%	23.5%	69.1%	49.8%	-51.3%	40.6%	19.3%
B	126.1%	-10.7%	62.1%	68.4%	46.1%	150.0%	-22.0%	72.0%
C	-20.0%	20.6%	-8.8%	17.7%	255.6%	-21.4%	10.0%	8.3%
D	0.0%	4.4%	0.0%	0.0%	0.0%	0.0%	0.0%	0.8%
E	0.0%	0.0%	0.0%	-0.1%	-0.4%	0.3%	0.0%	-0.2%
F	2.2%	35.5%	-76.7%	-30.8%	10.8%	-28.2%	23.3%	-4.7%
G	0.7%	0.0%	0.3%	0.0%	-0.7%	6.4%	0.0%	0.8%
I	38.5%	-2.3%	92.3%	185.7%	0.0%	-37.5%	0.0%	50.0%
J	74.4%	12.5%	32.5%	10.8%	0.0%	75.0%	5.0%	39.9%
K	10.5%	0.0%	2.6%	149.5%	2.7%	143.7%	0.0%	27.3%
L	0.0%	0.0%	0.0%	110.0%	16.6%	-85.5%	-22.8%	11.2%

We have used "Quartile.INC" function in Microsoft Excel (Microsoft, 2024) to calculate QTL_1 and QTL_3. It includes both the minimum and maximum values in the computation. This approach more suitable for project management datasets and for applied research, and is aligned with the boxplot definition by Tukey (Tukey, 1977). It takes cares of the full variability of the dataset as well as enable common comparability across all the functions.

As per Tukey rule (Tukey, 1977), for exploratory data analysis and consider mild outliers, following formula are used for limit calculation:

Lower Limit = QTL_1 - 1.5 (Inter Quartile Range)

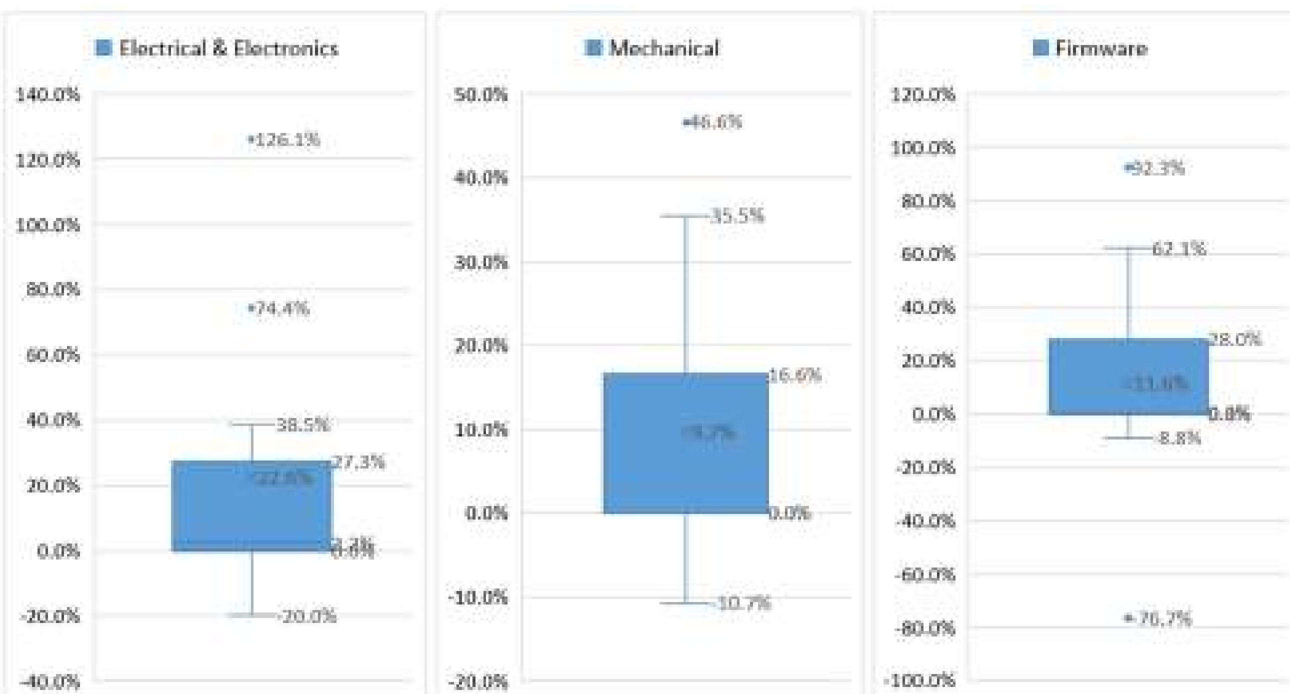
Upper Limit = QTL_3 + 1.5 (Inter Quartile Range)

The data points are considered to be outliers out of these limits.

Table 4.2 IQR calculation and Basic Statistics of the Project Data

Project	Electrical/ Electronics	Mechanical	Firmware	Software	SQA/ SW Testing	Other (PM, EM)	Tech Comm	Total
Min Value	-20.0%	-10.7%	-76.7%	-30.8%	-0.7%	-85.5%	-22.8%	-4.7%
QTL_1	0.0%	0.0%	0.0%	0.0%	0.0%	-32.8%	0.0%	0.8%
Median QTL_2	2.2%	0.0%	0.3%	17.7%	2.7%	0.0%	0.0%	11.2%
QTL_3	27.3%	16.6%	28.0%	89.5%	31.3%	40.7%	7.5%	33.6%
Max Value	126.1%	46.6%	92.3%	185.7%	255.6%	150.0%	40.6%	72.0%
IQR	27.3%	16.6%	28.0%	89.5%	31.3%	73.6%	7.5%	32.7%
Lower Limit	-40.9%	-24.8%	-42.0%	-134.3%	-47.0%	-143.2%	-11.3%	-48.3%
Upper Limit	68.2%	41.4%	70.0%	223.8%	78.3%	151.1%	18.8%	82.7%
Average Variance	23%	10%	12%	53%	35%	14%	3%	20%
Total Variance	248%	107%	128%	580%	380%	152%	34%	225%

Based on the Inter-Quartile Range (IQR), median, minimum and maximum values calculation, we have plotted the Box Plots as shown in figure 4.6.



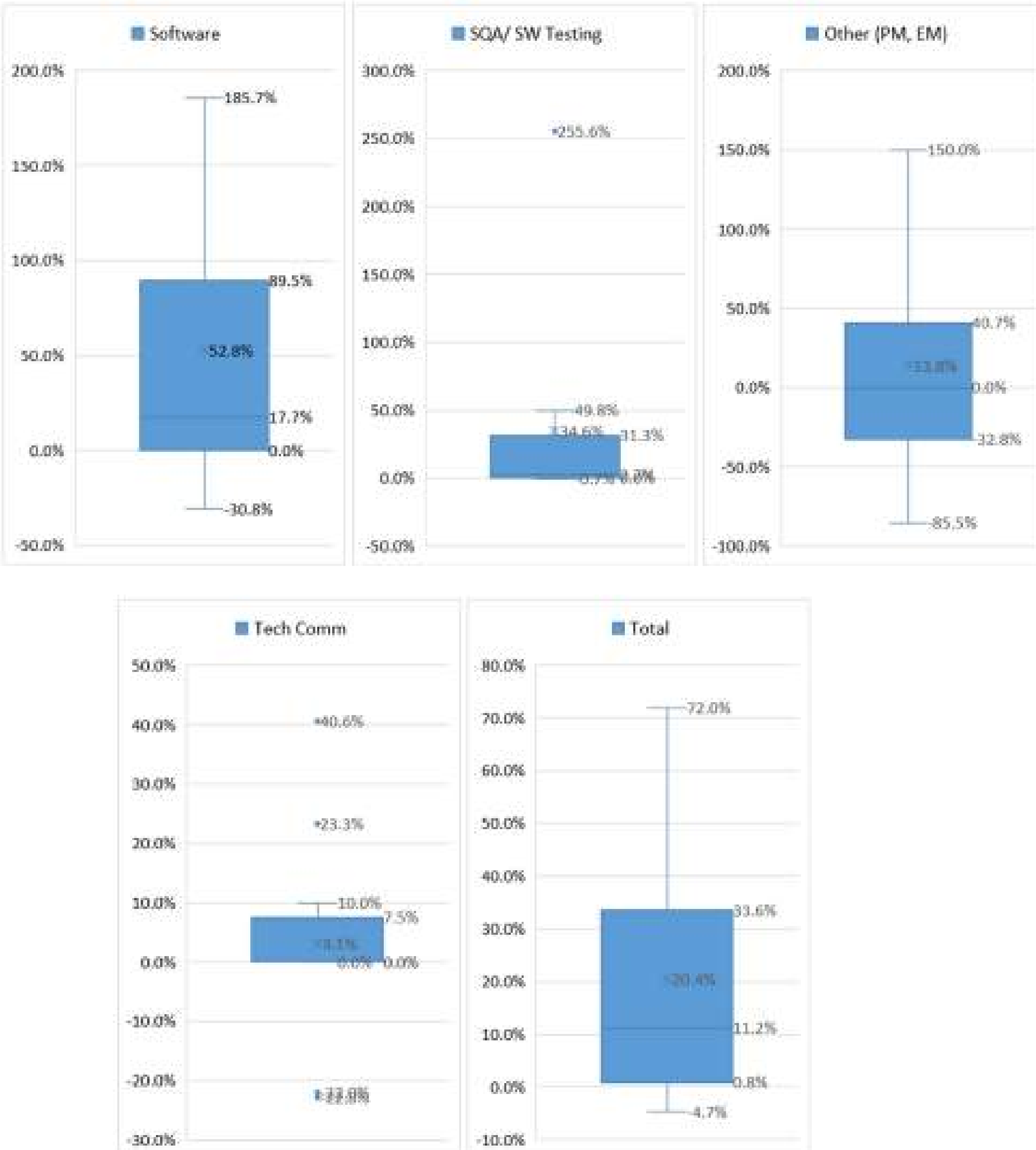


Figure 4.6 Functional Box Plots of Effort Variance data

Overall, the summation of project is indicated by Total value box plot, which shows very high variability, although the effort variance value in 20.4%. The Software testing function has effort variance value very high, but the spread of the data is not very high, which may be due to consistency in miscalculation of efforts or there was always higher efforts on testing due to research and development nature of these projects. The E&E (Electrical and Electronics) team also has a slightly higher spread of the data with 23% of the absolute effort variance value. This may be due to complexities in design of embedded electronics systems, chip design and geopolitical issues in raw material sourcing such as silicon for electronic items proto development and testing.

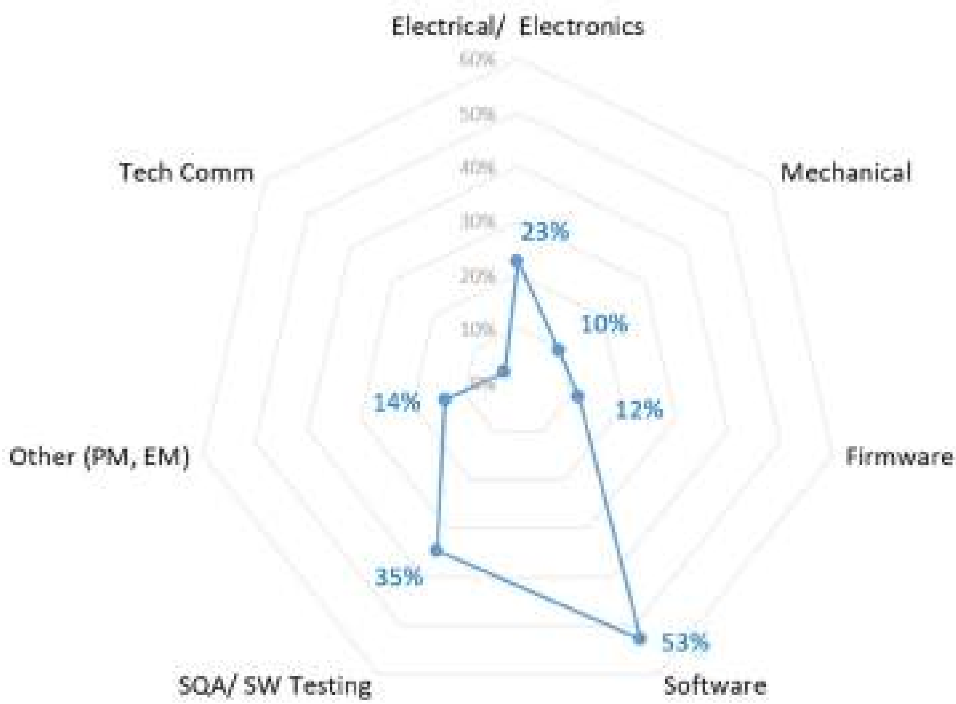


Figure 4.7 Average variance of all project

The Software Development function has very high spread of the data as well as the highest absolute effort variance. It may be due to iterative nature of software development process, frequently changing customer demands, code rework, scope addition, or even the team's capability and sluggish learning curve on newt technologies.

The function wise average effort variance is shown in form of spider chart in figure 4.7. It clearly indicates that, at the individual project level, the highest contributor to project success uncertainty are Software with 53% of cases and Software Testing functions with 35% cases across projects.

The cumulative efforts variance of all projects, which indicates performance at the portfolio level, is show in form of funnel chart in figure 4.8. It indicates the similar observation, i.e. Software development and Software Testing are the top two function which contributes the efforts variance.

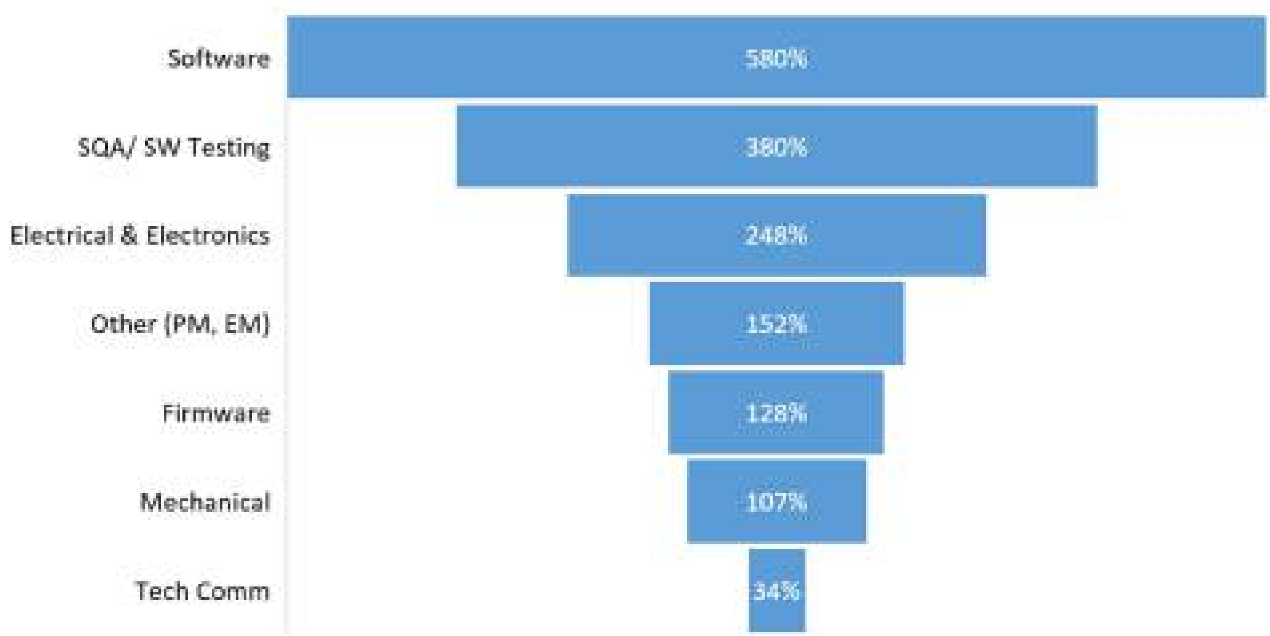


Figure 4.8 Cumulative variance of all projects

From the analysis of above data (Table 4.2) and Boxplots (Figure 4.6), Spider chart (Figure 4.7) and Funnel chart (figure 4.8), following are the insights (Refer table 4.3 for each functional area of the project:

Table 4.3 Insights from functional analysis of project data

Function	**Interpretation from Box Plots**
Electrical/ Electronics	Most of the projects delivered close to the plan except occasional overrun, Ex. +126%
Mechanical	No major issues. Well planned and predictable delivery. This is due to the perfect scope definition at the start of the work.
Firmware	Most of the projects balanced out the analysis, however occasional high underestimation, Ex. -76%
Software	Least predictable area. Highest spread amongst all functions due to requirement change & scope creep. This is highest risk area and needs most focus for the successful project delivery.
SQA/ SW Testing	Manageable for most of the projects, but few projects have high variance (Ex. +256.6%) due to unpredictability in re-testing needs, regression etc. Need to track closely and work on achieving higher accuracy in estimation.
Other (PM, EM)	High variability as this competency depends on project complexity and shared resources.
Tech Comm	Very much stable, predictable and well estimated. This is due to the similarity of technical communication and documentation work across all projects.
Total Portfolio	Overall, at portfolio level, projects fall within deviation. Software and Software testing are needs most attention for success of individual project as well as for portfolio success.

4.5 Industry Data – Schedule Variance Analysis and Insights

For the 11 projects, we collected milestone schedule data, quantified the delay in weeks and analyzed to categories the delay. Figure 4.9 shows that for each project; there are defined milestones and defined categories of delay.

Project
Project- A
Project- B
Project- C
Project- D
Project- E
Project- F
Project- G
Project- H
Project- I
Project- J
Project- K

Milestones
Engineering Release
Alpha Proto Build
Alpha Proto Test
Beta Proto Build
Beta Proto Test
Final Proto Build
Performance Test
Verification & Validation Test
Productionize

Delay Category
Development Resource
Development Test
Engineering Release
External Lab
MVP Build
Production
Project on Hold
Prototype
Quality Issue
Regulatory Test
Requirement Addition
Rework

Figure 4.9 Project, Milestones and Delay Categorization

The delay value in weeks for each milestone was bifurcated into different categories and based on analysis a “Delay Category” was assigned to each delay value.

After data collection and analysis, a box-plot is plotted for the delay categories considering the all the delay values as shown in figure 4.10 below. Other than, project on hold category, majority of the deviation and unpredictability is observed in Prototype build, Testing, Quality issues and Requirement Engineering. These are interrelated areas and need to be worked by project team to increase the predictability of the product delivery.

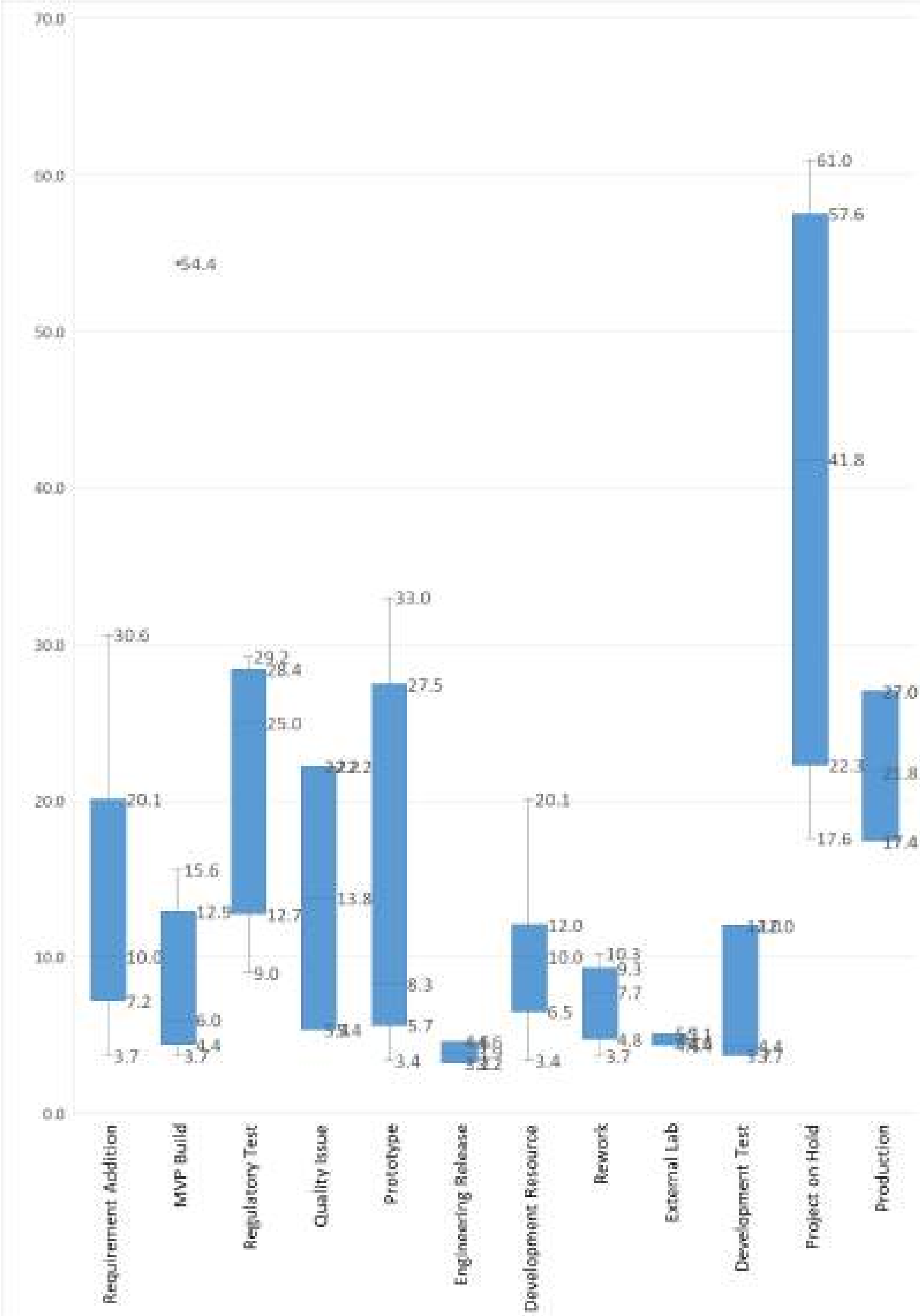


Figure 4.10 Box plot for delay categories

We have also conducted Pareto analysis (Figure 4.11) on the delay categorized data. We have eliminated ‘Project on Hold’ category as it was a management strategic decision and considering its magnitude, it may skew the analysis. The Development resource shifting is occasional and is a tactical decision. Major contributor for schedule variance is Requirement addition. Other factors and prototype and MVP build and testing. This proves that, for a software program to be successful, the team need to focus on Requirement Engineering, early proto and MVP build and testing to receive early feedback on the product.

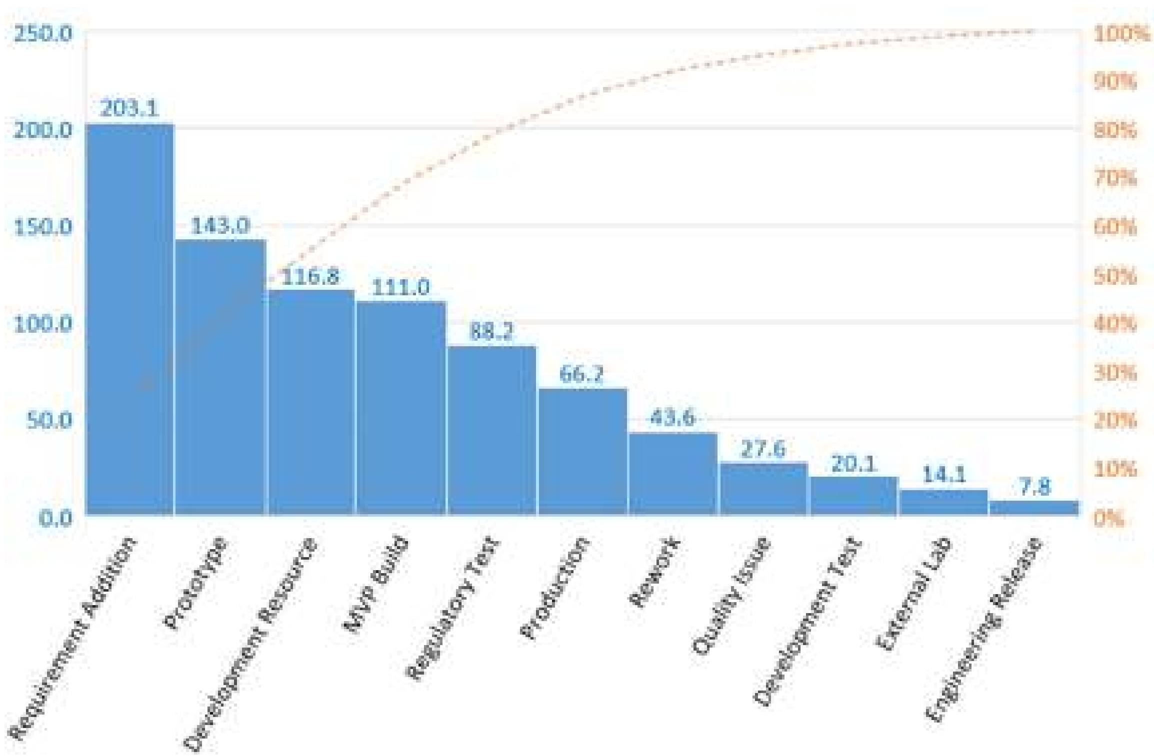


Figure 4.11 Schedule Variance Delay Category Pareto Chart Analysis

4.6 Insights from the study and Variable Definition

- Software Development and SQA (Software Quality Assurance/ Software Testing) functions has most unpredictability on efforts and schedule. So, these are the driving functions and has most activities on critical path of project schedule. For a success of a high-tech product development project, the focus should be on managing risks of Software and SQA area.
- Software requirement understanding, early focus on iteratively building MVP (Minimum Viable Product) and it's testing to get the early feedback is critical for predictable on-time launch.
- Hybrid product development approach is the most used methodology in the industry as it balances and inclusive of best of agile and waterfall methodologies. As the product becomes more and more hardware oriented, classic waterfall methodology is used. Whereas, as the product uses more and more software and new generation technologies such as AI, ML, GenAI etc., the product development process approaches more agility.
- The various project performance parameters are inter-related and impact the customer satisfaction. Effort and Schedule variance are importance parameters to be measured on a project which are inclusive of impact of the scope creep and quality issue/ rework, and they impact the cost variance. All variance such as Schedule, Effort, Cost, Scope and customer satisfaction. These relationships are shown in figure 4.12 below.

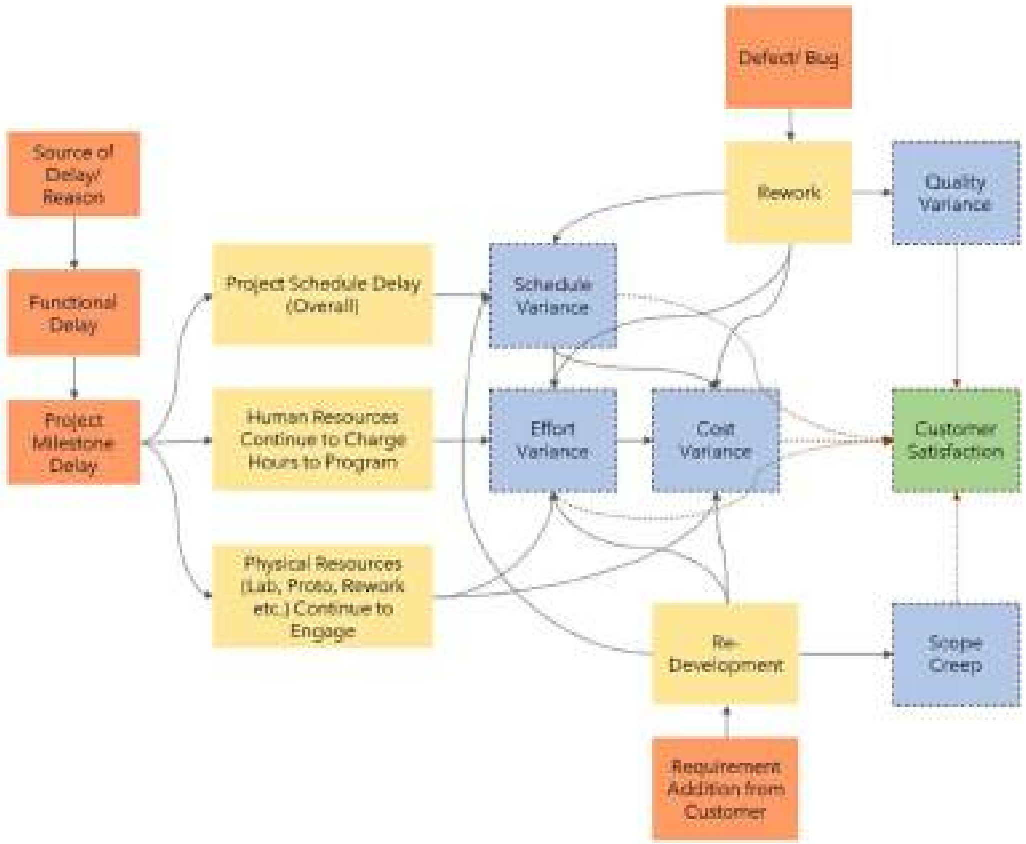


Figure 4.12 Industry Data – Schedule Delay Analysis and Insights

- We got a high-level relationship between governing parameters for project success from figure 4.12. Accordingly, we defined five dependent variables as shown in Table 4.4 below.

Table 4.4 Defining Dependent Variables

#	**Governing Parameter**	**Metric**	**Dependent Variable (y_i)**
1	Schedule Variable	Time based metric	y_1 = On-time Delivery
2	Cost Variance	Cost based metric	y_2 = No Budget over-run
3	Scope Creep	Change management-based metric	y_3 = No Scope creep
4	Quality Variance	Quality based metric	y_4 = No Quality defects
5	Customer Complaint	Customer experience-based metric	y_5 = High Customer/ stakeholder satisfaction

- Based on the insights on pitfalls from detailed analysis of literature, we have derived factors and from each factor we have defined the independent variables as shown in table 4.5 below.

Table 4.5 Defining Independent Variables

<table>
<tr><th>#</th><th>Reference</th><th>Insight on Pitfall</th><th>Factor</th><th>Independent variable (x_i)</th></tr>
<tr><td rowspan="2">1</td><td>Aulet, 2013; Tobak, 2014</td><td>Lack of deep understanding of customer problems</td><td rowspan="2">Empathy</td><td rowspan="2">x_1 = Clarity on Customer Requirements</td></tr>
<tr><td>Griffith, 2014; Ries, 2011; Blank, 2013; Christensen, 1997; Cooper, 2019</td><td>Solving a wrong problem</td></tr>
<tr><td rowspan="2">2</td><td>Burdakov et al. 2025; Walch, 2020; Singla et al., 2019; PMI, 2022</td><td>Non-suitability of common agile model to new generation technologies needing speed, automation & reliability</td><td rowspan="2">Automation</td><td rowspan="2">x_2 = Project Process Automation</td></tr>
<tr><td>Kettunen and Laanti, 2008; Brown, 2009</td><td>Linear & sequential integrated model, Rigid application</td></tr>
<tr><td rowspan="2">3</td><td>Serrador and Pinto, 2015; PMI, 2017; Morris, 2013; Cohn, 2004</td><td>Insufficient scope definition & Product vision - Scope Creep and Frequent prioritization</td><td rowspan="2">Vision</td><td rowspan="2">x_3 = Clarity on Product Vision</td></tr>
<tr><td>Kerzner, 2017; Crawford and Bryce, 2003; Christenson & Walker, 2008; Buchan et al., 2021</td><td>Lack of Vision - Improper definition of project key performance indicators (KPIs)</td></tr>
<tr><td>4</td><td>Frauenberger et al., 2017; Goodman et al., 2012</td><td>Cultural, ethical, ecological and sustainability aspects are often overlooked</td><td>Sustainability, Ethics</td><td>x_4 = Sustainability & Ethics Strategy</td></tr>
<tr><td>5</td><td>Ximenes, 2015; Jarke et al., 2011; Sawhney et al., 2005</td><td>Requirements do not come from real user</td><td>Requirement Source</td><td>x_5 = Requirement Source (Customer vs Internal)</td></tr>
</table>

#	Reference	Insight on Pitfall	Factor	Independent variable (x_i)
6	Deeb, 2013; Keller and Kotler, 2009	Lack on the right GTM (Go-To-Market) Strategy	Growth	x_6 = GTM, Growth & Scale-Up Strategy
	Dobrigkeit & De Paula, 2017; Vilkki 2010; Grossman-Kahn and Rosensweig 2012	Growth and increasing scalability post product launch		
	Kalenda et al., 2018	Insufficient Emphasis on Implementation – Execution and scaling		
7	Lukasik and Saylor, 2018; Ferreira, 2011	Missing the human aspect in the design	Human	x_7 = Human-Centered Design
	Dybå and Dingsøyr, 2008; Dobrigkeit et. al, 2019a; Curcio et al., 2019; Aldave et al., 2019	Lacks focus on design		
8	Boehm, 2011; Molokken and Jorgensen, 2003	Lack of awareness of the end-to-end product obligations	Scope	x_8 = Detailed Scope Definition
	Molamphy, 2025; Lukasik & Saylor, 2018	Impact of iterative nature on customer experience, by focusing on increments rather than overall scope		
9	Ramesh et al., 2010; Inayat et al., 2015; Fitriani et. al., 2016	Lack of sufficient documentation in agile	Governance	x_9 = Program Governance Strength
	Dikert et al., 2016; Goulstone, N.D.; PMI, 2017; Müller, 2009	Slipshod program governance		
10	Filippov et al., 2012; Iamratanakul et al., 2008	Lack of strategic alignment of project portfolio	Portfolio	x_{10} = Portfolio Strategy Management
11	Pai et al., 2021; Silva et al, 2017	Lack of formal issue/ defect management process	Issue/Defect	x_{11} = Issue/Defect Management

#	Reference	Insight on Pitfall	Factor	Independent variable (x_i)
12	Beckman & Barry, 2007; Brown and Katz, 2011; Paasivaara et al., 2018	Limited Feedback Loops	Feedback	x_{12} = Lessons Learnt Feedback Loop
13	Lindberg et. al., 2011	Barriers in information flow from DT phase to subsequent agile phase	Impediments	x_{13} = Bottlenecks & Impediments Resolution
	Wölbling et. al.' 2012	Teams' reluctance and organizational resistance		
14	Boehm and Turner, 2003; Pikkarainen et al., 2008	Fragmented non-integrated outcome	Integration	x_{14} = Project/ Product Integration Management
	Hossain et al., 2011; Stray et al., 2018	Lack of collaboration between various agile teams		
	Kolko, 2018; Meadows, 2008; Buchanan, 1992; Johansson-Sköldberg et al., 2013	Lack of system thinking		

4.7 Variable Relationship & Hypothesis Formation

The main dependent variable Y is a function of sub-dependent variable y_i

$Y = f(y_i)$

$Y = f(y_1, y_2, y_3, y_4, y_5)$ ---------------- (Function 1)

Where,

Y = Dependent Variable i.e. Project Success

y_i = Sub-dependent variable

Also, each of the sub-dependent variable y_i is a function of independent variable x_i

$y_i = f(x_i)$

$y_i = f(x_1, x_2, x_3 \ldots\ldots x_{14})$ ---------------- (Function 2)

Where,

x_i = Independent variable

Hence, from function 1 and 2, we get

$Y = f(x_i)$

Sub-Dependent variables (y_i) are (From Table 4.4):

y_1 = On-time Delivery (Time based metric)

y_2 = No Budget over-run (Cost based metric)

y_3 = No Scope creep (Change management-based metric)

y_4 = No Quality defects (Quality based metric)

y_5 = High Customer/ stakeholder satisfaction (Customer experience-based metric)

Independent variables (x_i) are (From Table 4.5):

x_1 = Clarity on Customer Requirements

x_2 = Project Process Automation

x_3 = Clarity on Product Vision

x_4 = Sustainability & Ethics Strategy

x_5 = Requirement Source (Customer vs Internal)

x_6 = GTM, Growth & Scale-Up Strategy

x_7 = Human-Centered Design

x_8 = Detailed Scope Definition

x_9 = Program Governance Strength

x_{10} = Portfolio Strategy Management

x_{11} = Issue/Defect Management

x_{12} = Lessons Learnt Feedback Loop

x_{13} = Bottlenecks & Impediments Resolution

x_{14} = Project/ Product Integration Management

Relationship between Dependent and Sub-dependent variable:

To set-up relationship between Y and y_i, we need to normalize the score. The y_1 to y_4 are lower the better, however, y_5 is higher the better.

Normalized scores (N_i) are:

$N_1 = 1 - (y_1/100)$

$N_2 = 1 - (y_2/100)$

$N_3 = 1 - (y_3/100)$

$N_4 = 1 - (y_4/100)$

$N_5 = y_5/10$

For N_i the value is higher the better for project success.

Hence, the Composite score for main dependent variable is:

$$Y = \frac{w_i.N_i}{\sum w_i}$$

$$Y = \frac{w_1.N_1 + w_2.N_2 + w_3.N_3 + w_4.N_4 + w_5.N_5}{w_1 + w_2 + w_3 + w_4 + w_5}$$

Where,

w_i are weightage to each sub-dependent variable.

So, now we have 1 derived dependent variable Y (Project Success) and 14 independent variables (x_i) as mentioned above.

Hypothesis:

We want to understand that whether there is any relationship between xi and Y and if yes, what is the degree of their relationship.

So, the hypotheses are as follows.

Table 4.6 Hypotheses

For Y = Project success, the independent variables x_i are as follows:

#	Independ-ent Variable	Independent Variable Description	Null Hypothesis (H_0)	Alternative Hypothesis (H_a)
1	x_1	Clarity on Customer Requirements	There is no significant relationship between Clarity on Customer Requirements and Project Success.	There is a significant relationship between Clarity on Customer Requirements and Project Success.
2	x_2	Project Process Automation	There is no significant relationship between Project Process Automation and Project Success.	There is a significant relationship between Project Process Automation and Project Success.
3	x_3	Clarity on Product Vision	There is no significant relationship between Clarity on Product Vision and Project Success.	There is a significant relationship between Clarity on Product Vision and Project Success.
4	x_4	Sustainability & Ethics Strategy	There is no significant relationship between Sustainability & Ethics Strategy and Project Success.	There is a significant relationship between Sustainability & Ethics Strategy and Project Success.
5	x_5	Requirement Source (Customer vs Internal)	There is no significant relationship between Requirement Source (Customer vs Internal) and Project Success.	There is a significant relationship between Requirement Source (Customer vs Internal) and Project Success.

#	Independent Variable	Independent Variable Description	Null Hypothesis (H_0)	Alternative Hypothesis (H_a)
6	x_6	GTM, Growth & Scale-Up Strategy	There is no significant relationship between GTM, Growth & Scale-Up Strategy and Project Success.	There is a significant relationship between GTM, Growth & Scale-Up Strategy and Project Success.
7	x_7	Human-Centered Design	There is no significant relationship between Human-Centered Design and Project Success.	There is a significant relationship between Human-Centered Design and Project Success.
8	x_8	Detailed Scope Definition	There is no significant relationship between Detailed Scope Definition and Project Success.	There is a significant relationship between Detailed Scope Definition and Project Success.
9	x_9	Program Governance Strength	There is no significant relationship between Program Governance Strength and Project Success.	There is a significant relationship between Program Governance Strength and Project Success.
10	x_{10}	Portfolio Strategy Management	There is no significant relationship between Portfolio Strategy Management and Project Success.	There is a significant relationship between Portfolio Strategy Management and Project Success.
11	x_{11}	Issue/Defect Management	There is no significant relationship between Issue/Defect Management and Project Success.	There is a significant relationship between Issue/Defect Management and Project Success
12	x_{12}	Lessons Learnt Feedback Loop	There is no significant relationship between Lessons Learnt Feedback Loop and Project Success.	There is a significant relationship between Lessons Learnt Feedback Loop and Project Success
13	x_{13}	Bottlenecks & Impediments Resolution	There is no significant relationship between Bottlenecks & Impediments Resolution and Project Success	There is a significant relationship between Bottlenecks & Impediments Resolution and Project Success.
14	x_{14}	Project/ Product Integration Management	There is no significant relationship between Project/ Product Integration Management and Project Success.	There is a significant relationship between Project/ Product Integration Management and Project Success.

Hypothesis and Variable Inter-relationship Conceptual Model:

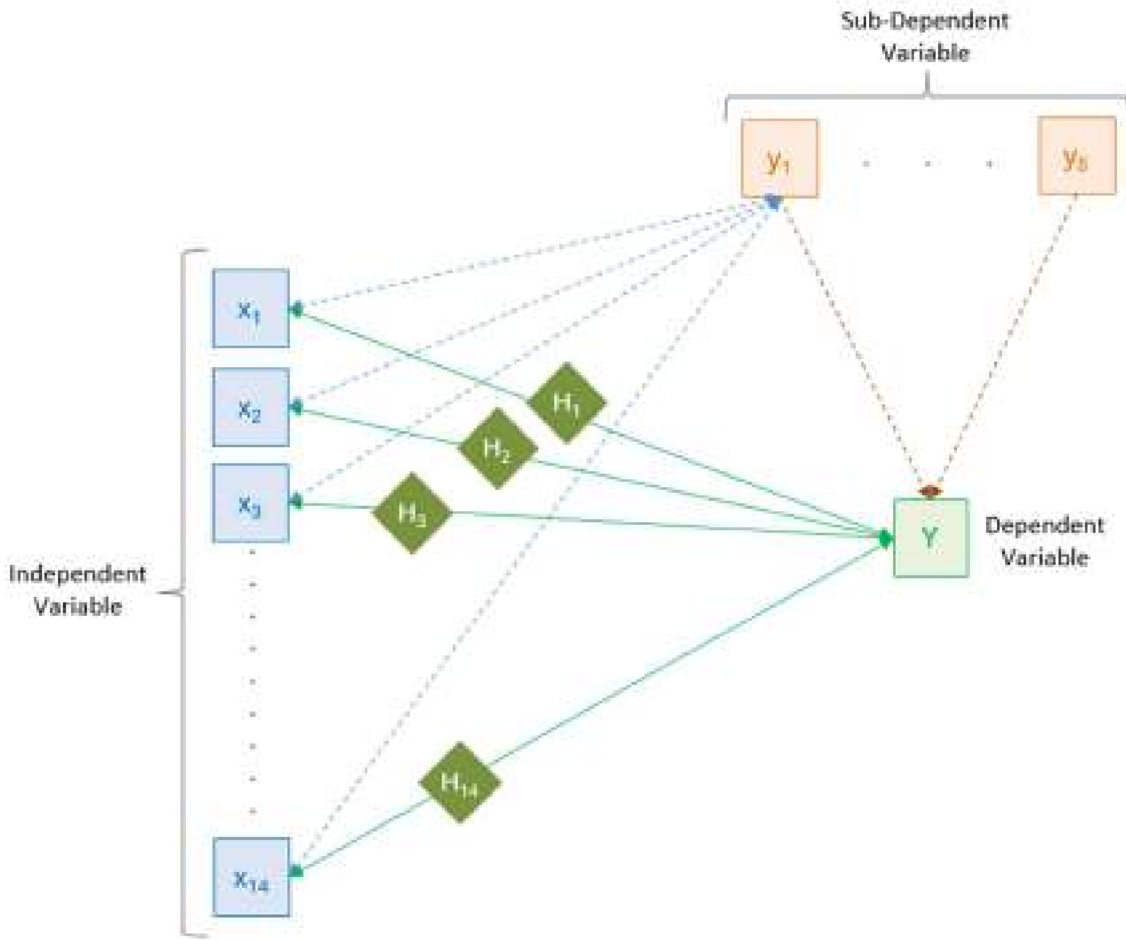


Figure 4.13 Hypothesis and Variable Inter-relationship Conceptual Model

Each individual independent variable (x_i) has effect on each sub-dependent variable (y_i). we are deriving the dependent variable (Y) using the composite score formula from the 5 sub-dependent variables (y_i). Hence to form hypothesis, the relation is formed between each independent variable (x_i) and the main dependent variable (Y). In this way, 14 hypotheses are formed for testing and analysis as shown in figure 4.13.

4.8 Survey – Questionnaire and Data Collection

Survey Questionnaire Form:

- We have created a question for each y_i and x_i. There 5 questions for y_1 to y_5 and 14 questions for x_1 to x_{14}).
- There is 1 question at the start to understand industry of the respondent.
- There is 1 optional open question at the end, where respondent can add any other factor which affect the project success.
- The 1 question on email ID of the respondent is kept optional to enable anonymity. Mandatory identification may prompt the respondent not choose to provide inputs.
- So, in total there are here are 23 questions in the survey.
- Questions to be used in survey are in **Blue Font** below.

What Makes Projects Succeed or Fail?

- A Survey on Analysis of Key Factors Affecting Project Outcome.

Please share your inputs from your experience about various factors responsible for inefficiencies in projects. Add average rating if you are working in multiple projects

1. **What type of Project you work on? (Example Industry: - Software, IT, Automobile, Banking, Consulting etc.)** ______________________________

Questions for dependent variable y_1 to y_5:

2. **What is % schedule delay on your program?** ______%

3. **What is % project cost over-run on your program?** ______%
4. **What is % time of total project duration spent on rework, fixing defects?** _____%
5. **What is % increase in scope (due to change requests, additional work, new features requested by customer post-delivery)?** ______%
6. **What is Customer/ Stakeholder satisfaction score on your project?** ______ On scale of 1 to 10 (1 being unhappy customer & 10 being customer delight)

7. PM Methodology adoption:

Which method(s) your organization use for project management? (Multiple Selection type question)

□ Phase-Gate/ Waterfall

□ Scrum

□ Scaled Agile (SAFe)

□ Hybrid (Agile + Waterfall)

□ Software Development Life Cycle (SDLC)

□ Design Thinking

□ System Thinking/ System Engineering

□ Lean Start-up

□ Other ______________________

Following questions are framed based on Improved Linkert scale as below:

○ **Strongly Agree**

○ **Agree**

○ **Somewhat Agree**

○ **Somewhat Disagree**

○ **Disagree**

○ **Strongly Disagree**

8. Clarity on Customer Requirements (x_1):

"The team deeply understands the customer's problem by empathizing on their needs before beginning of development work"

9. Project Process Automation (x_2):

"The project management/ development process is highly automated"

10. Clarity on Product Vision (x_3):

"The team has full clarity on the product vision even before the development starts, including what needs to be built and how will it solve the customer's exact problems."

11. Sustainability & Ethics Strategy (x_4):

"The project plan includes clearly defined steps to address sustainability and ethical considerations."

12. Requirement Source (x_5):

"Project requirements are received from direct interaction with customers or end-users rather than relying on internal team members such as product owners, product manager, business analysts, or marketing."

13. GTM, Growth & Scale-up Strategy (x_6):

"At the start of the project, the team clearly defines the plan for Go-To-Market (GTM), Growth, and business Scale-up in the future."

14. Human-Centered Design (x_7):

"While designing the product or service, there is a strong focus on Usability & Human aspects to make the product more user friendly."

15. Detailed Scope Definition (x_8):

"The project scope is fully defined and documented in detail before start of the development, with clearly stated project success criteria"

16. Strength of Program Governance (x_9):

"There is a formal program governance process for planning, execution, and tracking, including risk and change management practices, templates, and team religiously follow the process."

17. Portfolio Strategy Management (x_{10}):

"The project is aligned with organizational goals and supported by documents such as value propositions, opportunity analysis, and strategic alignment records."

18. Issue/Defect Management (x_{11}):

"Whenever there are quality issues found in testing and reported by customer, the team proactively follows a defined process to detect, track, analyze, and eliminate root causes of quality issues."

19. Lessons Learnt Feedback Loop (x_{12}):

"The retrospectives and lessons learnt reviews are effective and the team effectively track actionable items and integrated back into the project."

20. Bottlenecks & Impediments Resolution process (x_{13}):

"The project team follow a well-defined formal process to identify and remove bottlenecks and impediments in a timely and effective manner."

21. Product/Project Integration Management (x_{14}):

"The team successfully integrates deliverables from multiple project teams to form a whole cohesive product for customer."

22. **Do you face any other challenges on your project which results into delay, cost overrun, quality issues, scope change etc.? Or Anything else you want to add.**

This is an optional question; however, it will be great if you add your thoughts.

____

____

23. **Your email ID (Optional)**

____

4.9 Data Coding and Rationality Analysis

The survey was conducted for more than two months and 302 data points were collected against the minimum sample size of 282. After collecting the data, the file was downloaded as .csv file which then converted to .xlsx file.

The data coding was then performed as follows:

For dependent variables y_1_Schedule, y_2_Cost, y_3_Quality, and y_4_Scope, following formula is used for coding the data:

$N_i = 1 - (y_i/100)$

Where, $i = 1, 2, 3, 4$

For Dependent variable y_5_Customer, following formula is used for coding:

$N_5 = y_5/10$

The independent variables from x_1 to x_{14} are recorded using improved six-point Linkert scale, so, they are coded using the criteria mentioned in table 4.7 below. We replaced the survey response with a score value in a particular sequence. For example, 'Strongly Agree' response was replaced by score value 6.

Table 4.7 Coding criteria for independent variable

Sequence of Replacement in Dataset	Survey Response	Score as a replacement value for corresponding survey response
1	Strongly Agree	6
2	Somewhat Agree	4
3	Strongly Disagree	1
4	Somewhat Disagree	3
5	Disagree	2
6	Agree	5

The file was then cleaned and sanity check was performed as explained below. The excel data file was read using Python script in Jupyter Notebook which is an open-source web application with Anaconda ecosystem environment. It was observed that, these resources for data analysis are extremely powerful and convenient as it has pre-installed libraries such as NumPy, Pandas, Matplotlib, Scikit-learn etc. and it provides access to advanced modelling in Artificial Intelligence and Machine Learning, which can further help to build advanced statistical models for our dataset.

Top and bottom rows are observed as shown in figure 4.14 which depicts the very high-level overview of the survey dataset. Here there was also an opportunity to verify the accuracy of data coding.

```
# Check top 5 record in the dataset
df.head()
```

	y1_Schedule	y2_Cost	y3_Quality	y4_Scope	y5_Customer	x1_Empathy	x2_Automation	x3_Vision	x4_
0	0.600	0.85	0.70	1.0	0.7	2	1	3	
1	0.800	0.80	0.90	0.9	1.0	6	6	6	
2	0.850	0.80	0.80	0.9	0.8	5	4	4	
3	0.750	0.75	0.70	0.8	0.8	5	4	5	
4	0.775	0.90	0.85	0.8	0.6	3	1	4	

5 rows × 21 columns

```
# Check bottom 5 record in the dataset
df.tail()
```

	y1_Schedule	y2_Cost	y3_Quality	y4_Scope	y5_Customer	x1_Empathy	x2_Automation	x3_Vision	x
297	0.97	0.99	0.98	0.99	0.9	6	6	6	
298	0.98	0.98	0.97	0.98	0.9	5	5	5	
299	0.20	0.55	0.68	0.68	0.4	2	2	5	
300	0.90	1.00	0.99	0.97	0.8	4	6	6	
301	0.65	0.60	0.60	0.80	0.3	2	3	2	

5 rows × 21 columns

Figure 4.14 Data set high level overview: Top & Bottom rows

There are 302 data points (rows) which are indexed from 0 to 301 and 21 variables (columns). y_1 to y_5 are float values, x_1 to x_{14} are integers and method & Industry are object type data types. We also checked for missing records and found that, there are no null values. There are also no duplicate records in the data. This dataset rationality analysis is show in figure 4.15 below.

```
# Check information of the dataframe
# To check what type data is there in each column (Data type)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 302 entries, 0 to 301
Data columns (total 21 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   y1_Schedule         302 non-null    float64
 1   y2_Cost             302 non-null    float64
 2   y3_Quality          302 non-null    float64
 3   y4_Scope            302 non-null    float64
 4   y5_Customer         302 non-null    float64
 5   x1_Empathy          302 non-null    int64
 6   x2_Automation       302 non-null    int64
 7   x3_Vision           302 non-null    int64
 8   x4_SustEthics       302 non-null    int64
 9   x5_ReqSource        302 non-null    int64
 10  x6_Growth           302 non-null    int64
 11  x7_Human            302 non-null    int64
 12  x8_Scope            302 non-null    int64
 13  x9_Governance       302 non-null    int64
 14  x10_Strategy        302 non-null    int64
 15  x11_IssueRes        302 non-null    int64
 16  x12_Retrospective   302 non-null    int64
 17  x13_Impediments     302 non-null    int64
 18  x14_Integration     302 non-null    int64
 19  Method              302 non-null    object
 20  Industry            302 non-null    object
dtypes: float64(5), int64(14), object(2)
memory usage: 49.7+ KB
```

```
# Check for any null values
df.isnull().sum()
```

```
y1_Schedule          0
y2_Cost              0
y3_Quality           0
y4_Scope             0
y5_Customer          0
x1_Empathy           0
x2_Automation        0
x3_Vision            0
x4_SustEthics        0
x5_ReqSource         0
x6_Growth            0
x7_Human             0
x8_Scope             0
x9_Governance        0
x10_Strategy         0
x11_IssueRes         0
x12_Retrospective    0
x13_Impediments      0
x14_Integration      0
Method               0
Industry             0
dtype: int64
```

```
Number of Categorical columns =
['Method', 'Industry']

 And Number of Numerical columns =
['y1_Schedule', 'y2_Cost', 'y3_Quality', 'y4_Scope', 'y5_Customer', 'x1_Empathy', 'x2_Automation',
3_Vision', 'x4_SustEthics', 'x5_ReqSource', 'x6_Growth', 'x7_Human', 'x8_Scope', 'x9_Governance',
0_Strategy', 'x11_IssueRes', 'x12_Retrospective', 'x13_Impediments', 'x14_Integration']
```

```
# Check for any duplicate data records in the dataset:
dups = df.duplicated()
print('Number of duplicate rows = %d' % (dups.sum()))

df[dups]
```

```
Number of duplicate rows = 0
```

y1_Schedule	y2_Cost	y3_Quality	y4_Scope	y5_Customer	x1_Empathy	x2_Automation	x3_Vision	x4_SustEthics	x5_ReqSource	...

0 rows × 21 columns

Figure 4.15 Dataset Rationality Analysis – Data points, Variables and Data Types; Null & Duplicate values check

4.10 Principal Component Analysis (PCA)

Principal Component Analysis (PCA), a statistical technique to reduce high number of dimensions (in our case, they are five, such as y_1 to y_5) to one lower dimension which can explain most of the variance in data, was performed on the dataset. Five principal components are created. Figure 4.16 shows contribution of each of the variable in each of the principal components. The contribution is also shown as heat map in figure 4.17.

PCA Loadings (Variable Contributions):

	PC1	PC2	PC3	PC4	PC5
y1_Schedule	0.455234	-0.081648	-0.413890	-0.401296	-0.673610
y2_Cost	0.450567	-0.381825	-0.374937	-0.183665	0.690571
y3_Quality	0.456363	-0.086662	-0.072784	0.869016	-0.154066
y4_Scope	0.428117	0.872598	0.054896	-0.096458	0.207293
y5_Customer	0.445196	-0.280367	0.824503	-0.201831	-0.051513

Explained Variance by Each Component:

	Principal Component	Explained Variance Ratio
0	PC1	0.920337
1	PC2	0.040445
2	PC3	0.020553
3	PC4	0.010210
4	PC5	0.008454

Figure 4.16 PCA Output and Loading Data

Based on variance analysis, as shown in Scree plot in figure 4.18, it is observed that, the first principal component PC1 explains 92% of variance which is well above threshold of 80%.

So, based on majority of contribution and explained variance, we have selected PC1 as Project Success dependent variable Y. We have defined this key variable as "Y_Project_Success" which represents the combined effect of y_1 to y_5.

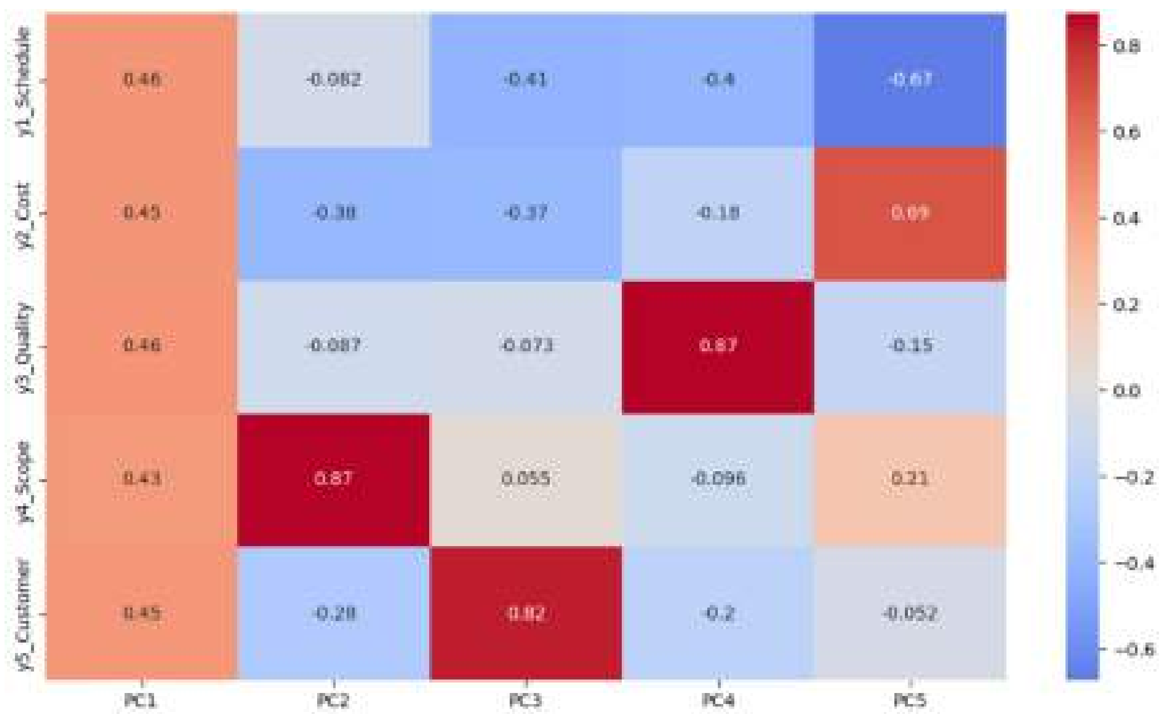


Figure 4.17 PCA Loading Heatmap

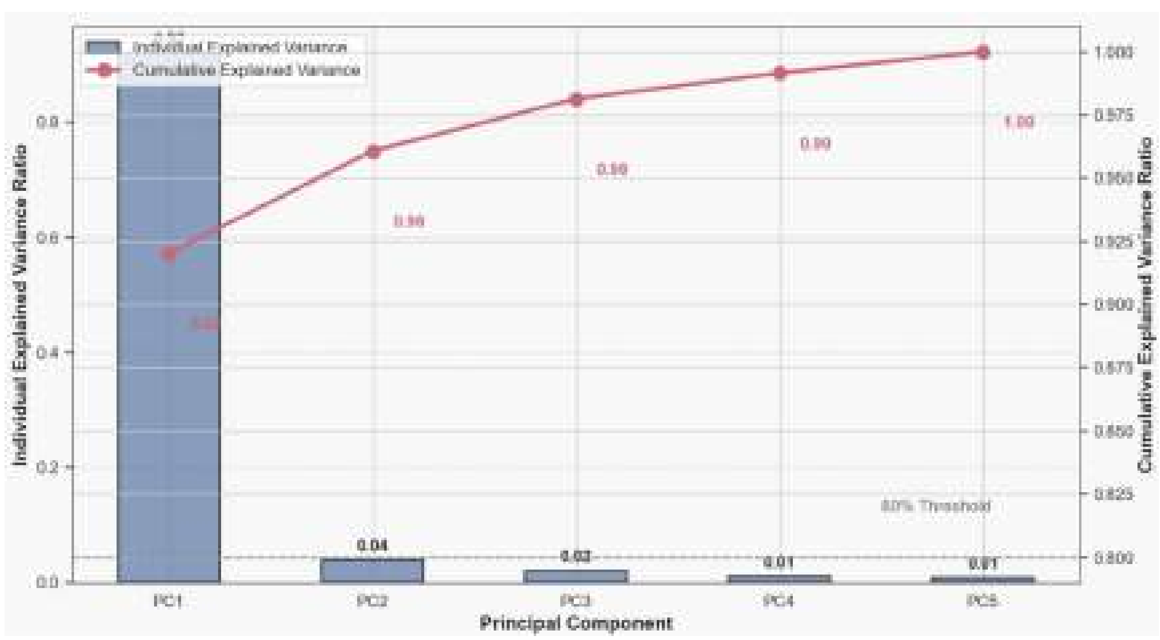


Figure 4.18 Scree Plot with Cumulative Explained Variance

4.11 Exploratory Data Analysis (EDA)

As a part of EDA, statistical summary is created as shown in figure 4.19 for Dependent Variables (y_i), Project Success variable (Y) and Independent Variable (x_i).

Table 4.8 Statistical Summary of Dataset

Variable	Mean	SD	Min	25%	50%	75%	Max	Range	IQR	Variance	COV
y1_Schedule	0.752	0.172	0.2	0.59	0.785	0.938	1	0.8	0.348	0.030	0.229
y2_Cost	0.773	0.172	0.5	0.6	0.82	0.95	1	0.5	0.350	0.030	0.223
y3_Quality	0.800	0.134	0.6	0.67	0.83	0.94	0.99	0.39	0.270	0.018	0.167
y4_Scope	0.816	0.135	0.5	0.693	0.82	0.95	1	0.5	0.258	0.018	0.166
y5_Customer	0.577	0.257	0.1	0.3	0.6	0.8	1	0.9	0.5	0.066	0.446
Y_Project_Success	-1.41E-16	2.14871	-3.571	-2.088	0.309	2.170	3.155	6.726	4.258	4.61697	-1.52E+16
x1_Empathy	3.493	1.688	1	2	4	5	6	5	3	2.849	0.483
x2_Automation	3.381	1.829	1	2	3	5	6	5	3	3.346	0.541
x3_Vision	3.781	1.578	1	2	4	5	6	5	3	2.490	0.417
x4_SustEthics	3.331	1.504	1	2	3	4	6	5	2	2.262	0.452
x5_ReqSource	3.447	1.445	1	2	3	5	6	5	3	2.089	0.419
x6_Growth	3.652	1.706	1	2	4	5	6	5	3	2.912	0.467
x7_Human	3.864	1.423	1	3	4	5	6	5	2	2.025	0.368
x8_Scope	3.570	1.481	1	2	4	5	6	5	3	2.193	0.415
x9_Governance	3.536	1.658	1	2	3	5	6	5	3	2.748	0.469
x10_Strategy	3.149	1.454	1	2	3	5	6	5	3	2.114	0.462
x11_IssueRes	3.646	1.493	1	3	4	5	6	5	2	2.230	0.410
x12_Retrospective	3.626	1.781	1	2	4	5	6	5	3	3.172	0.491
x13_Impediments	3.262	1.528	1	2	3	5	6	5	3	2.333	0.468
x14_Integration	3.460	1.650	1	2	3	5	6	5	3	2.721	0.477

Based on the complete dataset, box-plots are also plotted for dependent variables y_i and Y as shown in figure 4.19 for independent variables x_i as shown in figure 4.20. It provides the spread of the data readings with mean and inter-quartile ranges (IQR). The project success (Y) parameter is balance without any major skewness. Except the customer satisfaction (y_5), all y_i are slightly skewed with marked values generally on higher side. There are no outliers on any variable as the data points are not widely spread based on participant's response.

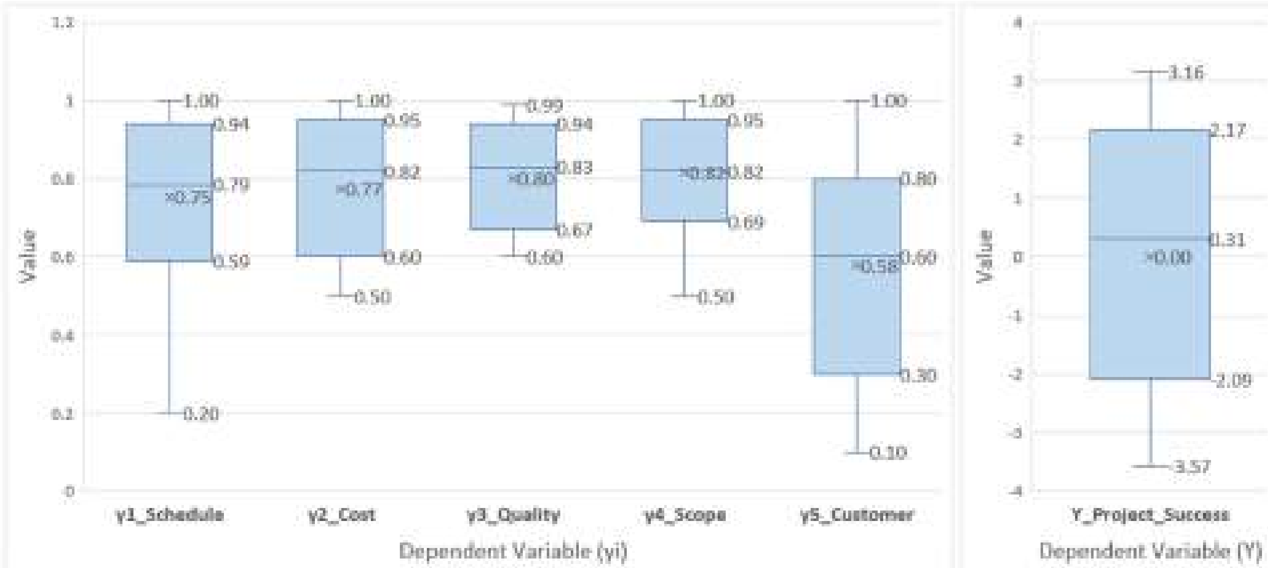


Figure 4.19 Box Plot for Dependent Variable (y_i)

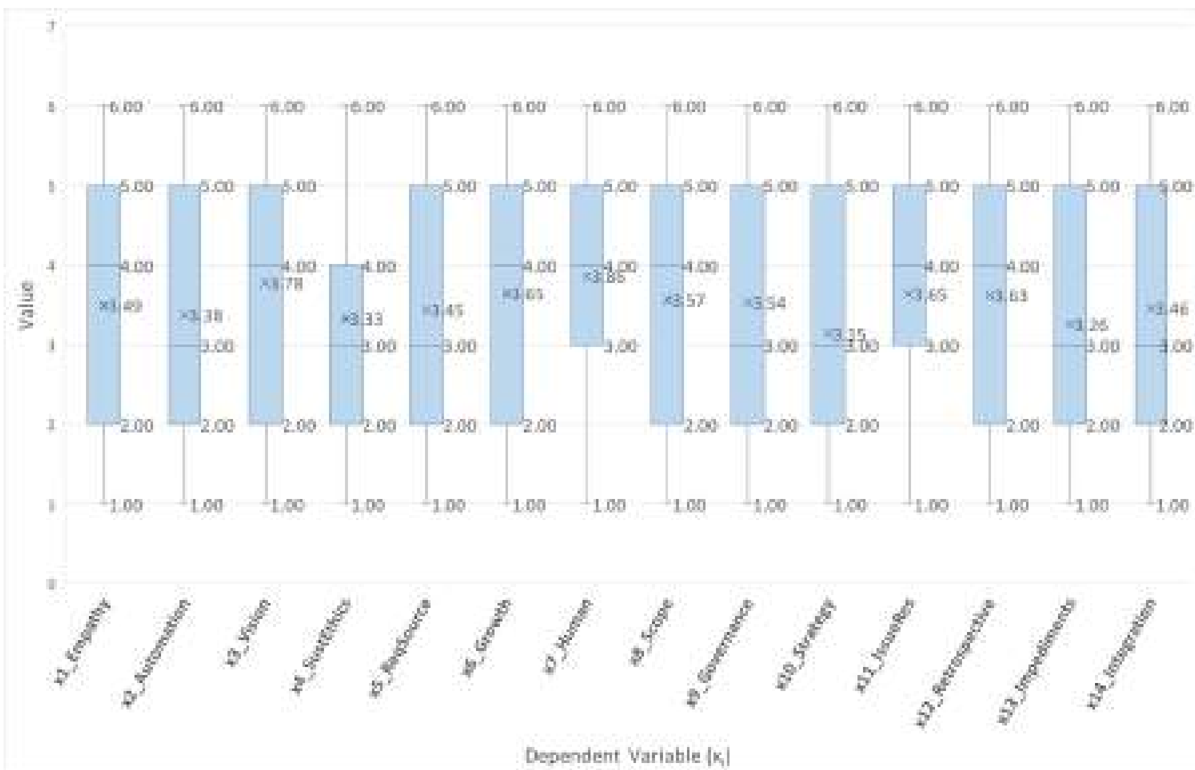
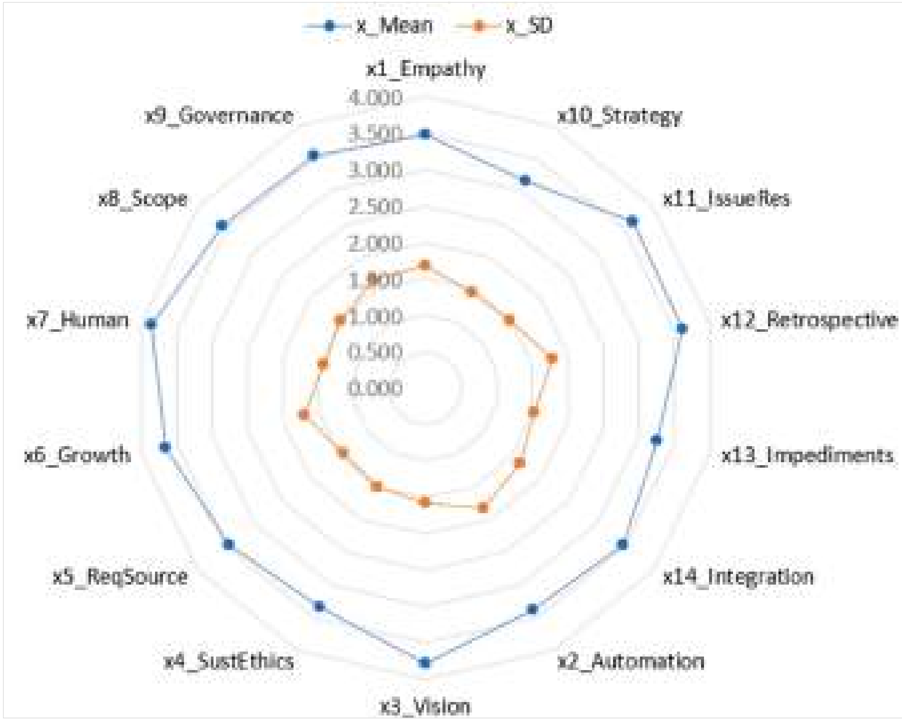


Figure 4.20 Box Plot for Independent Variable (x_i)



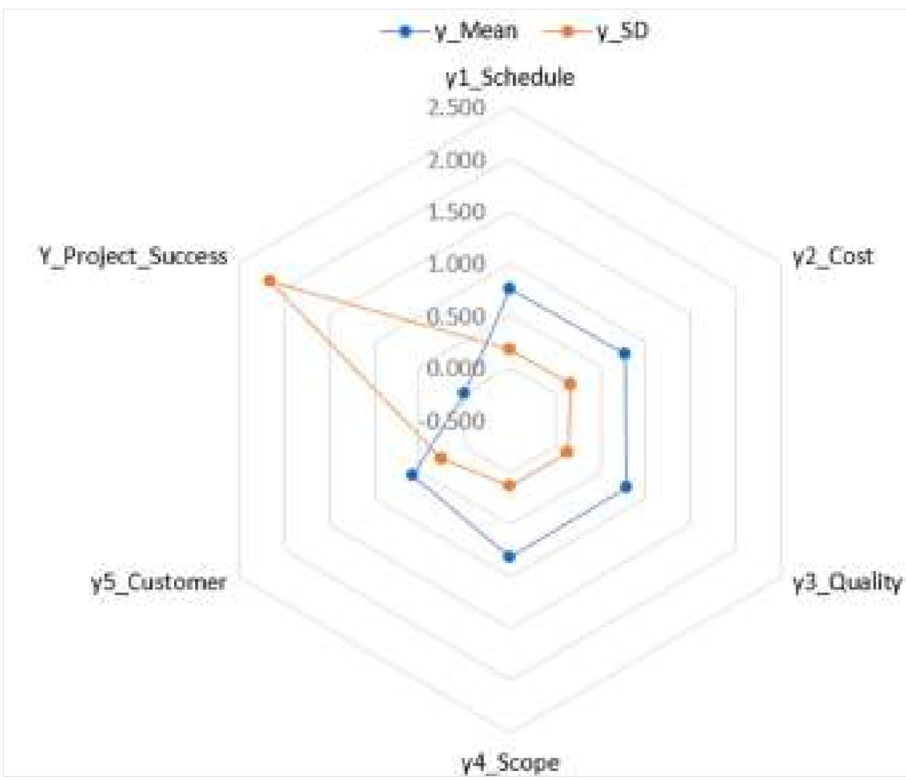


Figure 4.21 Variable mean & Standard Deviation radar

The mean and standard deviation are plotted in form of radar diagram for all variables as shown on figure 4.21. This provides better visualization for clear comparison between variables in terms of their effect on project success and consistency in results.

Sustainability & Ethics (x_4), Human aspects in product design (x_7) and Issue resolution process (x_{11}) are mores consistently adopted as the spread of the data is narrow. The mean for mean for project success (Y) parameter is 0.31 which is close to zero, which makes the data more balanced. This is mainly because, the Y parameter is derived from Principal Component Analysis and contains both positive and negative values.

Insights from initial EDA:

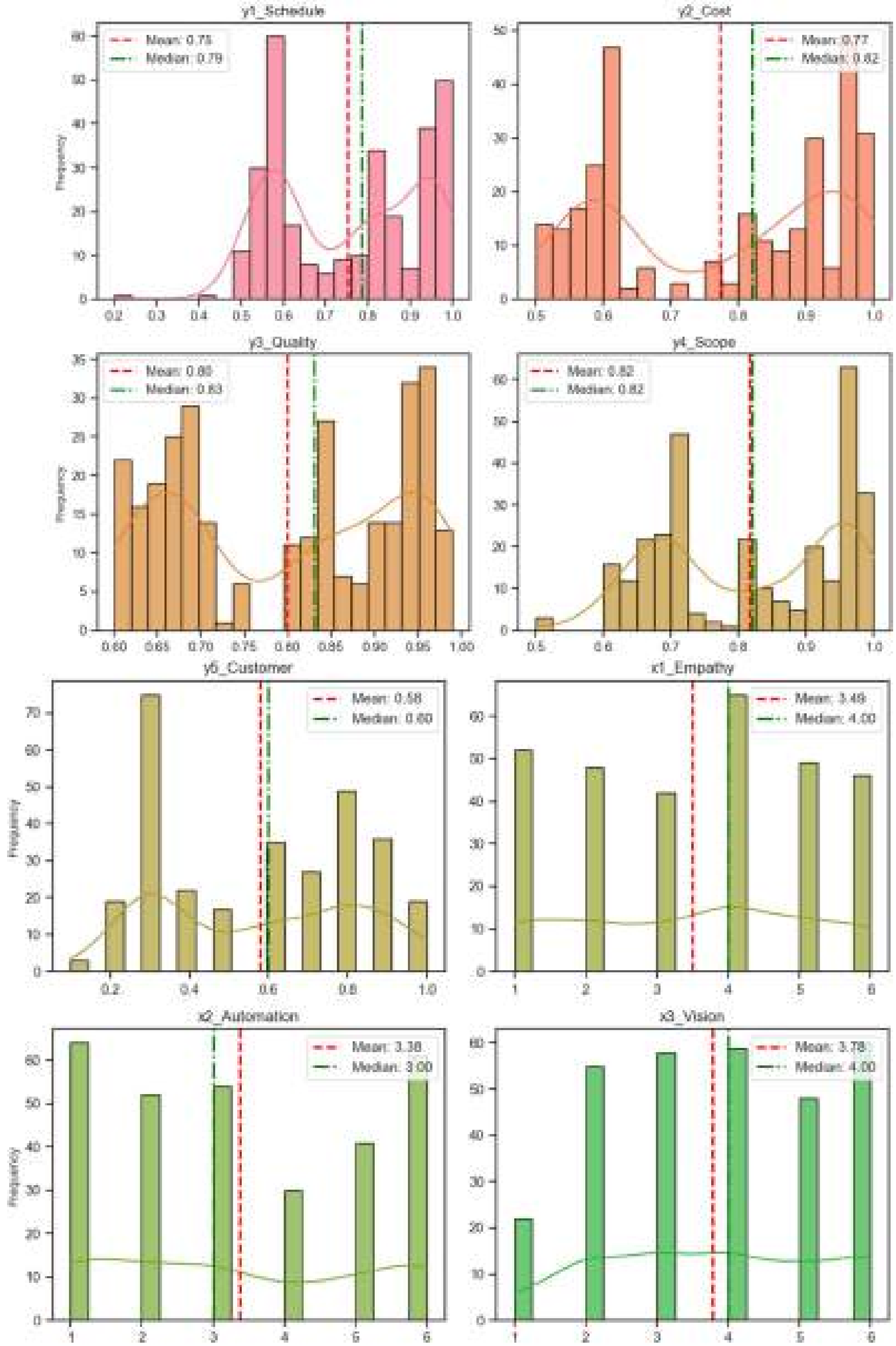
The dependent variables Schedule and Cost are consistent with low variability (Standard deviation 0.172) and mean 0.752 and 0.773 respectively. Quality (mean 0.8) and Scope (mean 0.816) have least standard deviation (SD) and higher means, which shows that the projects were able deliver on scope and quality most of the times. Customer satisfaction has highest variance with standard deviation as 0.257 and coefficient of variance (COV) as 0.446, and least mean of 0.577. This means that, the project were able to deliver well on internal success parameters of scope, quality, time and cost, however the customer may not have been consistently satisfied. So, building a right product on time with right quality and cost is an important aspect of project, however building what is values by customer need to be taken care in projects. This is also evident from the mean, SD and COV values of overall project success Y.

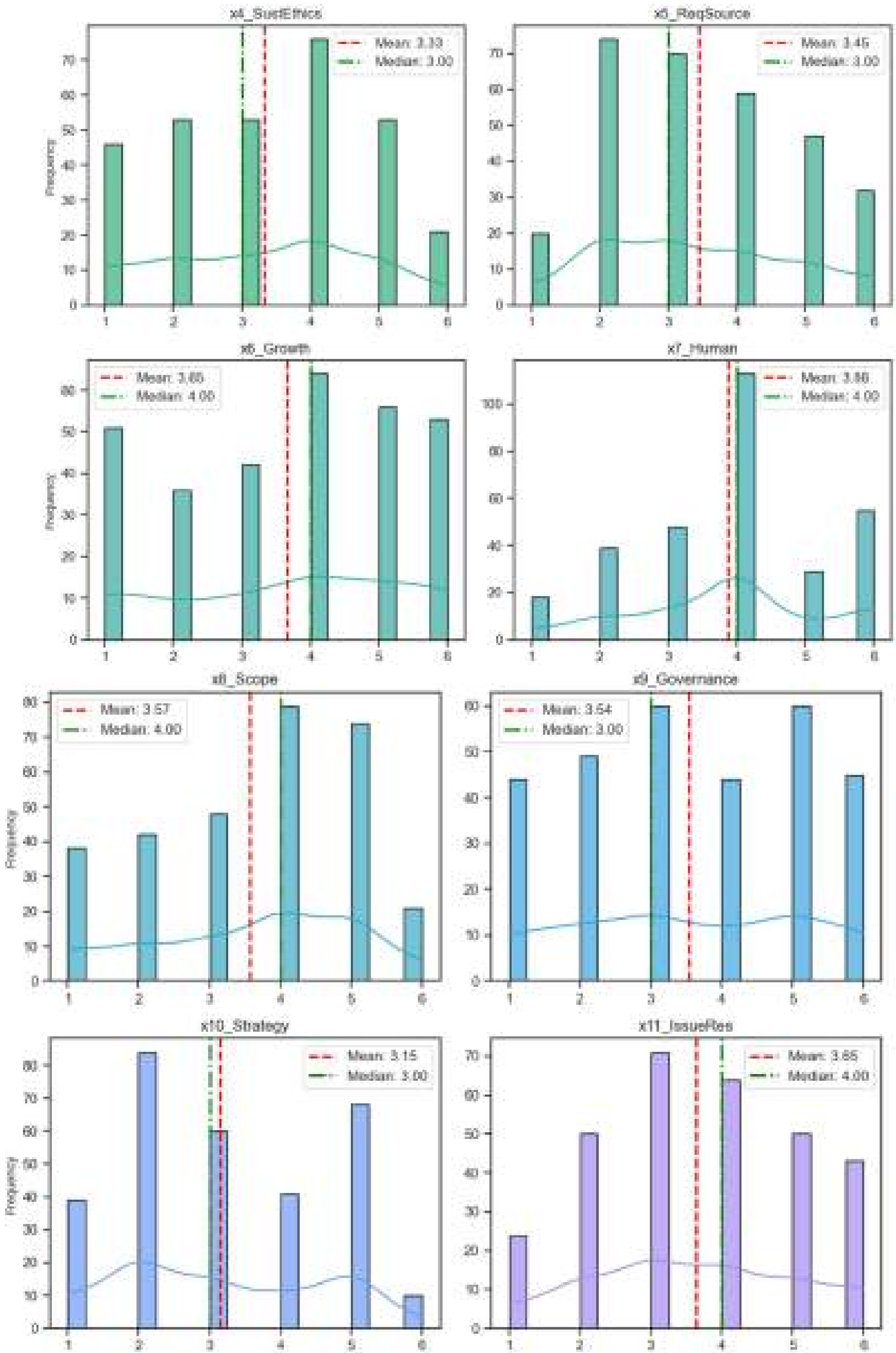
On independent variables (x_i), the COV range is 0.36 to 0.54, which is quite narrow. The higher COV means more inconsistent performance. Human centered design (x7) and vision (x3) are strongest factors with least variance & SD and highest mean. Sustainability & Ethics, Strategy, impediments removal and Automation are relatively weak with slightly higher COV and least mean. The factors with relatively highest variation in data are Governance, Empathy, Growth, Retrospective and Automation. So, it means that, majority of project success is driven by focus on Human centered design (x7) and clarity in customer requirements (x5) and product vision (x3).

We can’t quantitatively establish a relationship between x_i and project success from this statistical analysis, as it will calls for a more advance modelling. However, it is clearly visible that, on these projects, the cultural strengths such as human collaboration and vision clarity are driving factors. Automation and retrospective feedback loop variance indicates high inconsistency, which means some project teams incorporates automation and integrate feedback from retrospective session effectively in the product development process, however other teams are weak on these aspects. The sustainability and Ethics also shows inconsistency and weakness on these projects, even though the focus on ESG (Environmental, Social, Governance) is increasing now a days. Overall, a higher focus on Empathy, Growth, Retrospective, Governance and Automation can yield better project success, especially customer satisfaction.

Histograms for all variables:

The histograms are plotted for each of the dependent and independent variables as shown in figure 4.22. The histogram with mean and median lines are helpful for deriving insights as it provide information on spread, symmetry and skewness of the dataset.


y1_Schedule
Mean: 0.75
Median: 0.79
Frequency
y2_Cost
Mean: 0.77
Median: 0.82
y3_Quality
Mean: 0.80
Median: 0.83
Frequency
y4_Scope
Mean: 0.82
Median: 0.82
y5_Customer
Mean: 0.58
Median: 0.60
Frequency
x1_Empathy
Mean: 3.49
Median: 4.00
x2_Automation
Mean: 3.38
Median: 3.00
Frequency
x3_Vision
Mean: 3.78
Median: 4.00


x4_SustEthics
Mean: 3.33
Median: 3.00
Frequency
x5_ReqSource
Mean: 3.45
Median: 3.00
x6_Growth
Mean: 3.65
Median: 4.00
Frequency
x7_Human
Mean: 3.86
Median: 4.00
x8_Scope
Mean: 3.57
Median: 4.00
Frequency
x9_Governance
Mean: 3.54
Median: 3.00
x10_Strategy
Mean: 3.15
Median: 3.00
Frequency
x11_IssueRes
Mean: 3.65
Median: 4.00

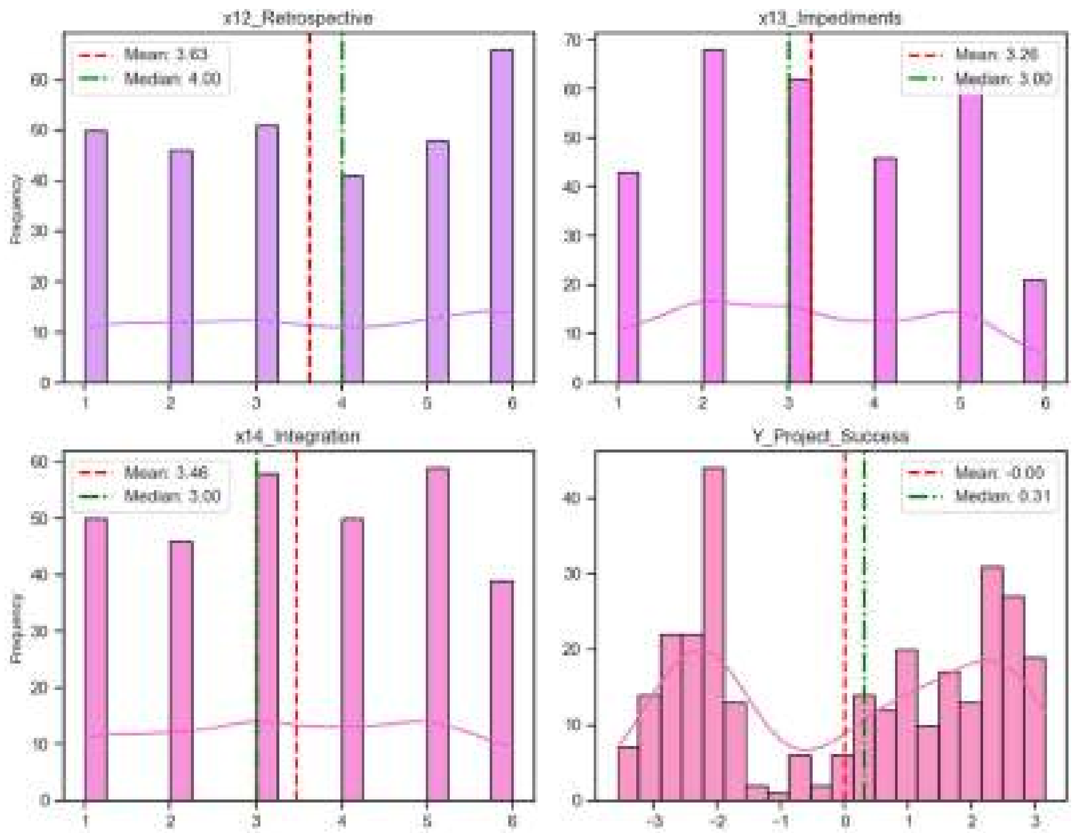


Figure 4.22 Histograms for all variables

For project Scope (y_4) and Customer satisfaction (y_5), the mean and median approximately coincides which indicates very minimal skewness and absence of outliers. For the other yi such as Schedule (y_1), Cost (y_2) and Quality (y_3), the median is greater than mean, which indicates the data to be slightly left skewed, i.e. a greater number of values toward the left, but overall clustered towered higher values of right-hand side. Also, in almost all cases, participant chose to provide values in higher or lower bracket, creating a clustered data, which also evident from multiple peaks of the histogram. This also because of the variation in level of governance, team maturity and style of project management across projects.

For independent variables such as Empathy (x_1), Automation (x_2), Growth (x_6), Governance (x_9), Retrospective (x_{12}) and Integration (x_{14}), the spread and the distribution of the data is fairly uniform which means there are no dominating values.

On Sustainability and Ethics (x_4), Requirement source (x_5), Strategy (x_{10}), impediment removal process (x_{13}) have mean greater than the mean, which means that, the data is skewed on the right side with tail of the curve on right hand side. So, higher number of responses are marked on the lower side of Linkert scale. Whereas, Clarity on product vision (x_3) and Scope definition robustness (x_8) have mild skewness toward left which suggests while many participants have responded as low to medium on these areas, but majority if the data points are towered very high ratings. The data is mostly normally distributed for Human-centered Design (x_7) and Issue resolution process definition (x_{11}) which indicate the right central tendency of the data.

Spearman Correlation Matrix:

The Spearman correlation values between various variables are calculated and plotted as a heat map as shown in figure 4.23. The matrix provides a very rich insights on relationship between variables. The dependent variables (y_i) are strongly correlated with project success (Y) with a range of correlation values from 0.92 to 0.98, which shows the effectiveness of the Principal Component Analysis (PCA).

The independent variables such as Impediment removal process (x_{13}), Issue resolution process (x_{11}), Strategy (x_{10}), Sustainability and Ethics (x_4) has low strength of correlation with project success variables y_i with a range of coefficient from 0.43 to 0.58. Whereas, Empathy (x_1), Automation (x_2), Growth (x_6), Human centered design (x_7), Retrospective feedback (x_{12}), vision clarity (x_3) and Requirement clarity (x_5) are strongly corelated to project success variable (Y) with range of values from 0.73 to 0.89. They are also strongly correlated to dependent variables (y_i) especially with Schedule, Customer satisfaction and Scope parameters.

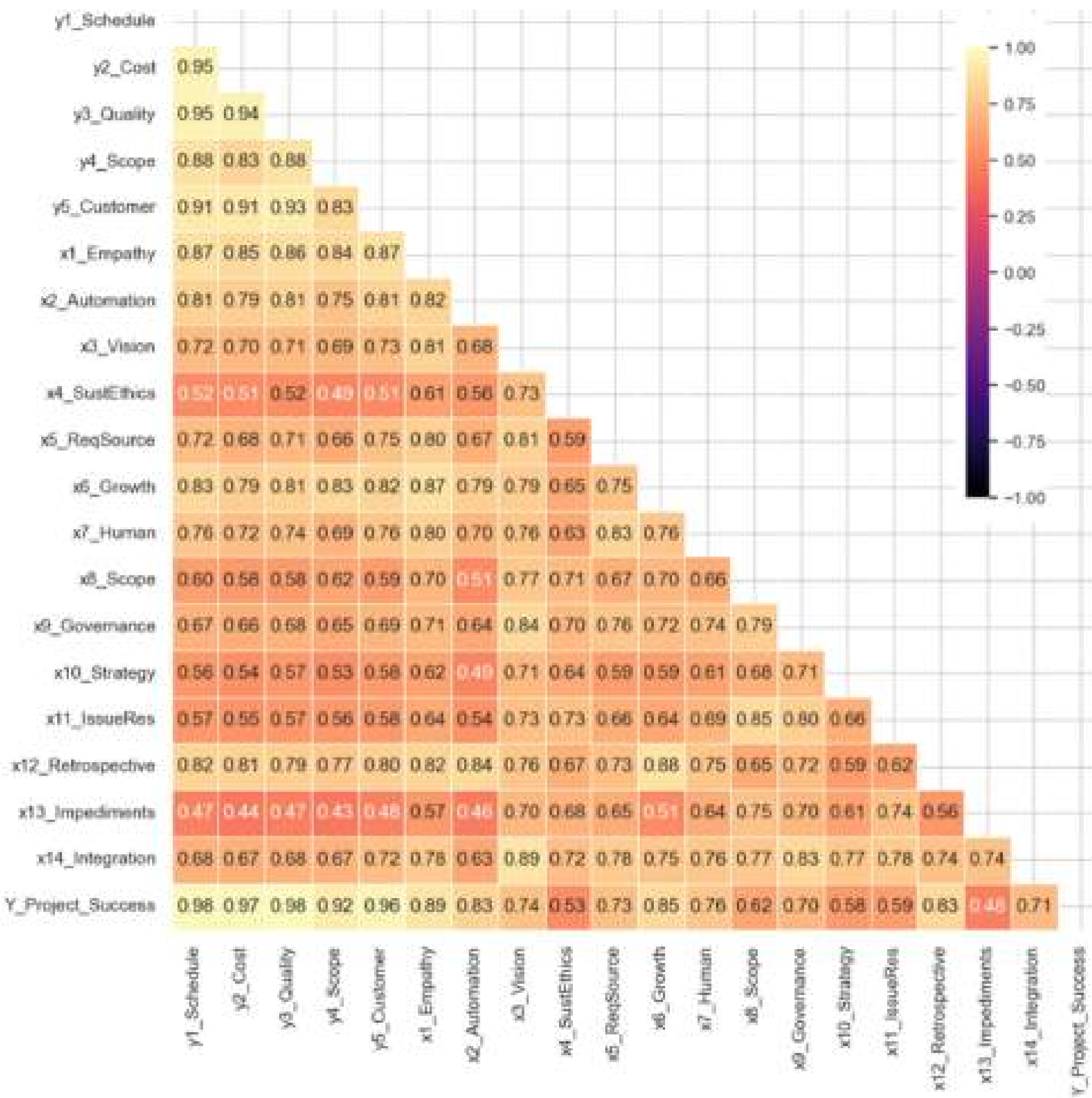


Figure 4.23 Spearman Correlation Matrix Heatmap

Reliability Check (Cronbach's Alpha)

To check the consistency of the data, Cronbach’s Alpha was calculated. For dependent variables (y_i) Cronbach’s Alphas values is found to be 0.961 and for independent variables(x_i) it is found to be 0.970. These values are greater than 0.90 which is benchmark for consistency (Mohd. Arof et.al., 2018), so the data consistency is excellent. This means that, the constructs for both dependent and independent variables are highly correlated and measures the similar underlying dimension, which indicates high statistical reliability. So, we can use the dataset for further analysis and modelling.

4.12 Multiple Linear Regression (MLR) Model

The Multiple Linear Regression was build using Python code for the dataset for one dependent variable Project Success (Y) and independent variables (from x_1 to x_{14}). We have used Ordinary Least Square (OLS) method to find the best fitting model. It is done by minimizing the sum of squared difference between predicted and observed values of x_i which are actually influenced by Y.

The output of the model development is shown in figure 4.24. The R^2 (0.859) and Adjusted R^2 (0.852) indicates that, 85% of the project success is explained by 14 variables (x_i), which means the MLR model is very strong. The F-Statistic of 124. 8 and p-Value < 0.05 indicate that the model is highly significant and predictors explain the project success.

The skewness (0.139) is close to zero, which indicates that the residuals are mostly symmetrically distributed around the mean value. Kurtosis (3.736) is close to 3, which is a value for normal distribution. Omnibus probability (0.05) suggest very marginal deviation from perfect normality. Jarque-Bera probability (0.0202) is less than threshold of 0,05, which shows the mild normality, which could be due to mild skewness to right hand side.

Durbin-Watson statistic value (1.656) is slightly less than 2, which shows very mild auto-correlation and perfectly acceptable considering the large dataset of cross-sectional data. The condition number 48.7 is well below 1000 and shows low to moderate multicollinearity which is not a serious concern.

Overall, the MLR model is valid and states the key significant variables governing the project success. It will also ensure the reliable estimation and valid hypothesis testing.

```
                            OLS Regression Results
==============================================================================
Dep. Variable:      Y_Project_Success   R-squared:                       0.859
Model:                            OLS   Adj. R-squared:                  0.852
Method:                 Least Squares   F-statistic:                     124.8
Date:                Thu, 02 Oct 2025   Prob (F-statistic):          4.91e-113
Time:                        16:49:46   Log-Likelihood:                -363.26
No. Observations:                 302   AIC:                             756.5
Df Residuals:                     287   BIC:                             812.2
Df Model:                          14
Covariance Type:            nonrobust
=====================================================================================
                        coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------
const                -4.1769      0.156    -26.750      0.000      -4.484      -3.870
x1_Empathy            0.6434      0.078      8.296      0.000       0.491       0.796
x2_Automation         0.1735      0.058      2.980      0.003       0.059       0.288
x3_Vision            -0.0382      0.083     -0.462      0.644      -0.201       0.124
x4_SustEthics        -0.2109      0.055     -3.826      0.000      -0.319      -0.102
x5_ReqSource         -0.0765      0.072     -1.069      0.286      -0.217       0.064
x6_Growth             0.2003      0.074      2.698      0.007       0.054       0.346
x7_Human              0.1449      0.071      2.039      0.042       0.005       0.285
x8_Scope             -0.0530      0.079     -0.674      0.501      -0.208       0.102
x9_Governance         0.1680      0.066      2.527      0.012       0.037       0.299
x10_Strategy          0.0706      0.054      1.310      0.191      -0.035       0.177
x11_IssueRes          0.0455      0.072      0.628      0.530      -0.097       0.188
x12_Retrospective     0.1924      0.068      2.839      0.005       0.059       0.326
x13_Impediments      -0.1028      0.058     -1.784      0.076      -0.216       0.011
x14_Integration       0.0018      0.079      0.023      0.982      -0.154       0.157
==============================================================================
Omnibus:                        5.993   Durbin-Watson:                   1.656
Prob(Omnibus):                  0.050   Jarque-Bera (JB):                7.800
Skew:                           0.139   Prob(JB):                       0.0202
Kurtosis:                       3.736   Cond. No.                         48.7
==============================================================================
```

Figure 4.24 Multiple Linear Regression result

From p-value and t-statistics analysis as shown in table 4.9, it is concluded that x1_Empathy ($p < 0.001$), x2_Automation ($p = 0.003$), x4_SustEthics ($p < 0.001$), x6_Growth ($p = 0.007$), x7_Human ($p = 0.042$), x9_Governance ($p = 0.012$), and x12_Retrospective ($p = 0.005$) are the most significant factors which contributes to project success. Considering the high t statistics, Empathy (x_1) is the highest contributing factor. This is because it is directly associated with deeply understanding the customer requirements, pain & gain. The Human centered design (x_7) is a borderline variable as it is slightly less than 0.05.

Table 4.9 MLR model – Significance Analysis

Variable	Coefficient	p-Value	|t| Statisctic
x1_Empathy	0.6434	0.000	8.296
x2_Automation	0.1735	0.003	2.98
x12_Retrospective	0.1924	0.005	2.839
x6_Growth	0.2003	0.007	2.698
x9_Governance	0.168	0.012	2.527
x4_SustEthics	-0.2109	0.000	3.826
x7_Human	0.1449	0.042	2.039
x13_Impediments	-0.1028	0.076	1.784
x10_Strategy	0.0706	0.191	1.31
x5_ReqSource	-0.0765	0.286	1.069
x8_Scope	-0.053	0.501	0.674
x11_IssueRes	0.0455	0.53	0.628
x3_Vision	-0.0382	0.644	0.462
x14_Integration	0.0018	0.982	0.023

Interestingly, out of all the significant variable, Sustainability & Ethics (x_4) has negative coefficient, stating the project success likely increases with decrease in efforts of on this variable. We need to consider & understand the contextual meaning for this case. The survey question for x_4 was about having dedicated steps in project management framework for sustainability & ethics. Generally, for many companies, these steps are defined as a part of policies such code of conduct policy. So, many participants given low rating on this variable, even though the project performed well. Also, on the other hand, although the sustainable and ethical practices are important from long term strategy point of view, it adds process, overheads, compliance complexity and resource constraints, which may reduce very short-term project efficiency.

Other parameters such as Vision (x3), Requirement Sourcing (x5), Scope (x8), Strategy (x10), Issue Resolution (x11), Impediments (x13), and Integration (x14) are not statistically significant ($p < 0.05$; $|t| < 2$). In practice, these variables may still be important, however, their unique contribution is limited and overshadowed by other significant parameters.

MLR Model Diagnostics:

We have trained and tested the dataset using Multiple Linear Regression Model, and found the R-squared and RMSE values as shown in table 4.10.

Table 4.10 MLR model – Performance Analysis

Metric	Training	Testing
R^2 (Coefficient of Determination)	0.845	0.91
RMSE (Root Mean Squared Error)	0.847	0.629

From R-Squared values, the tested model explains 91% variability in the data with no overfitting. And from RMSE value, we get a small error value of 0.629. This indicates excellent model performance with high prediction accuracy.

We also plotted the Residuals vs Fitted Values plot as shown in figure 4.25. The scatter plot is mostly random; however, the red LOESS (Locally Estimated Scatterplot Smoothing) curve shows a wavy pattern, which suggest a non-linear trend.

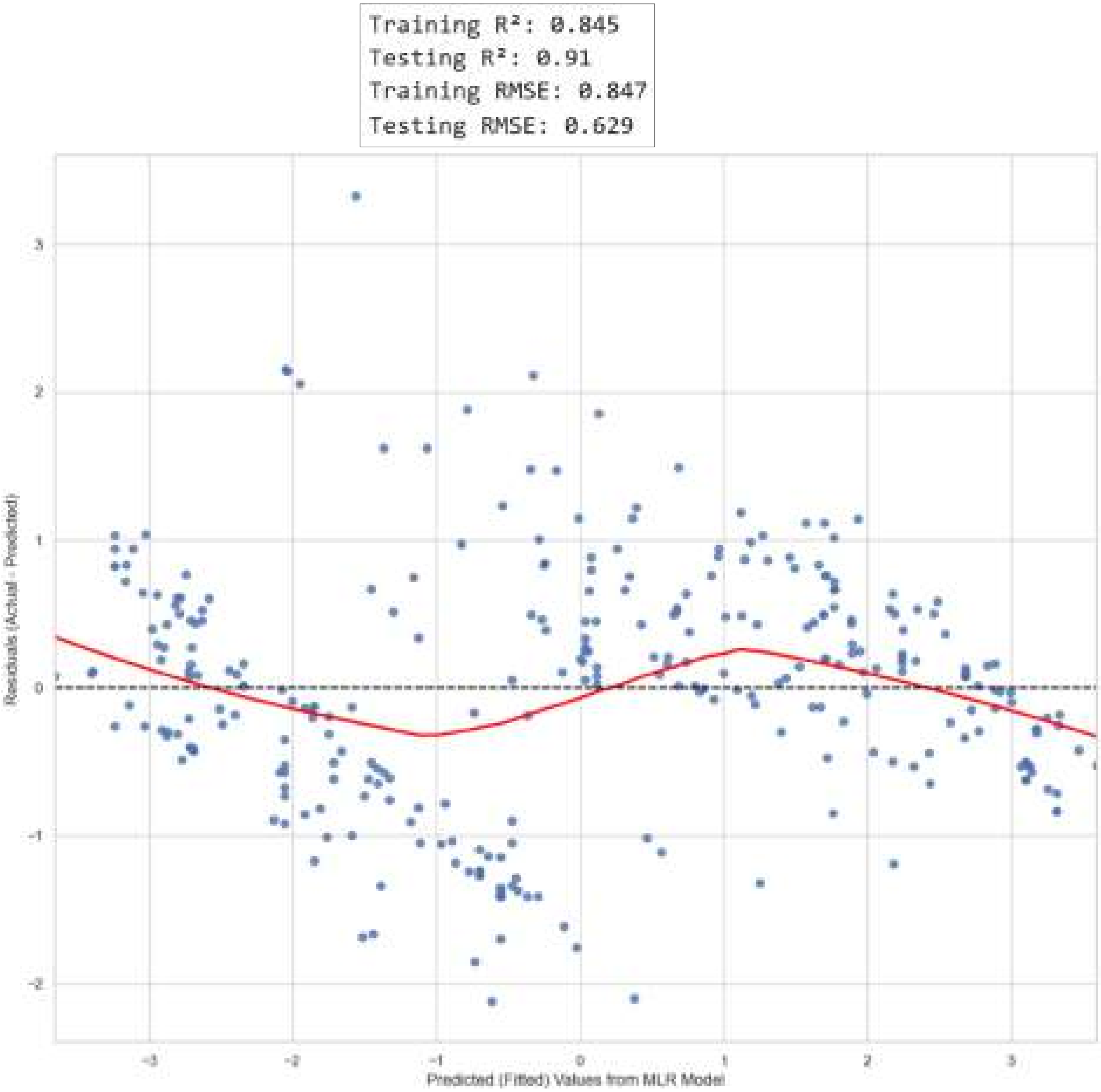


Figure 4.25 Residuals vs Fitted Values plot for MLR Model

So, due to potential presence of non-linearity, we will also build a non-linear model such as GAM (Generalized Additive Model) model which has capability to capture the smooth non-linear relationship between x_i and Y.

We have also calculated VIF (Variance Inflation Factor) to calculate multi-collinearity as shown in figure 4.26 below.

	Feature	VIF
0	const	10.780676
1	x1_Empathy	7.551236
2	x2_Automation	4.997084
3	x3_Vision	7.489544
4	x4_SustEthics	3.029832
5	x5_ReqSource	4.714035
6	x6_Growth	7.075942
7	x7_Human	4.508693
8	x8_Scope	5.968890
9	x9_Governance	5.348959
10	x10_Strategy	2.706961
11	x11_IssueRes	5.159629
12	x12_Retrospective	6.420119
13	x13_Impediments	3.413066
14	x14_Integration	7.496404

Figure 4.26 Multi-Collinearity Check using VIF (Variance Inflation Factor)

Few variables have VIF score, which indicates very low multi-collinearity. Other variables have VIF score between 5 to 8, which indicates moderate multi-collinearity. There is no major issue of multi-collinearity as no variable VIF score is above 10. To dela with moderate multi-collinearity, we will be building LASSO (Least Absolute Shrinkage and Selection Operator) regression model.

4.13 GAM (Generalized Additive Model)

To introduce the non-linearity between the independent variables (x_i) and dependent variable (Y Project Success), we have built GAM and observed output as shown in 4.27.

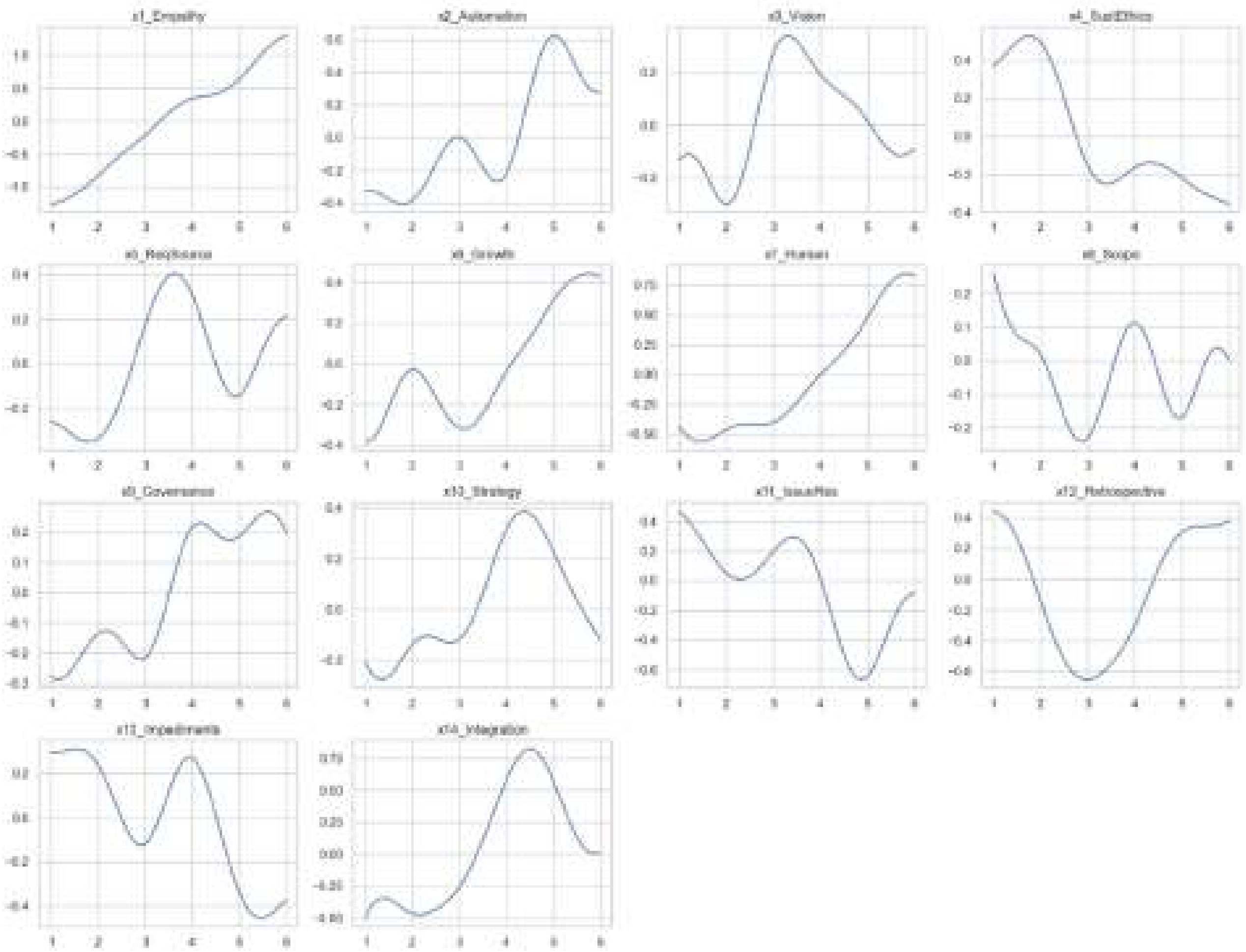


Figure 4.27 GAM (Generalized Additive Model) – Partial Dependence Plot Output

The partial dependance plots show the relationship between each independent variable (xi) with project success Y in isolated manner. x1_Empathy shows a strong positive linear relationship, whereas x4_Sustainability and Ethics has declining negative relationship with Y. x7_Human, x6_Growth, x2_Automation and x3_Vision have non-linear but positive relationship with project success. They specificity with respect to Y such as Automation picks up at the top where as vision picks up till mid but decline after that. There are oscillating curves with no specific conclusive relationship for x8_Scope and x13_Impediments. The U-shaped curve for x12_Retrospective shows that the mid-level practices may not yield a great project success. In this way, the GAM model helps to capture complex non-linear relationship which is not possible for regular OLS models.

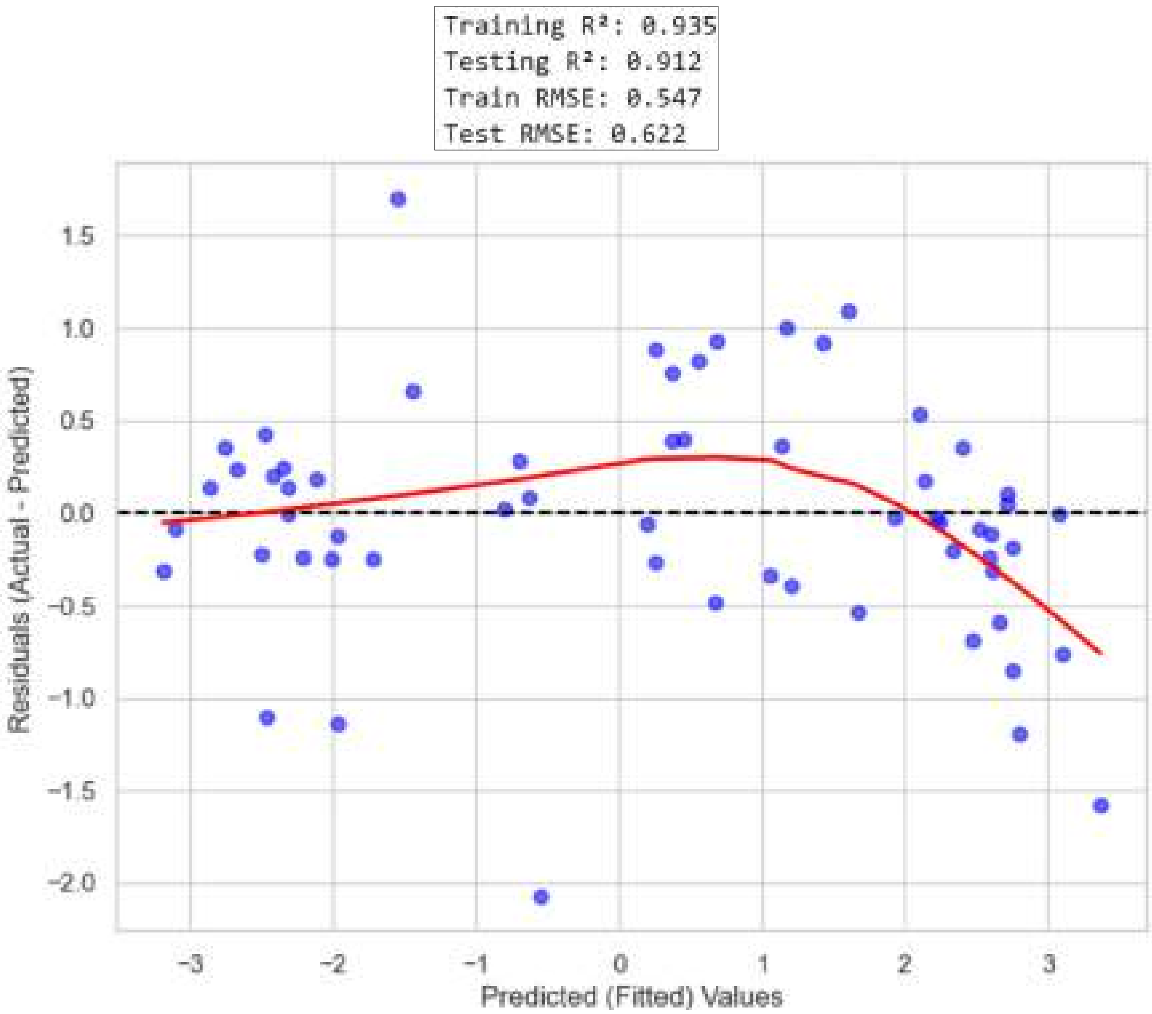


Figure 4.28 Residuals vs Fitted Values plot for GAM model

To analyze the performance of the GAM, we have calculated R2 and RMSC score in table 4.11, and residual vs fitted plot value as shown in figure 4.28.

Table 4.11 GAM Model – Performance Analysis

Metric	Training	Testing
R^2 (Coefficient of Determination)	0.935	0.912
RMSE (Root Mean Squared Error)	0.547	0.622

The GAM explains 93.5 % (R-Squared) of variation in the data. For testing, it is 91.2 % on unseen data, which shows a strong generalization of the model. The test RMSE for training and testing data are 0.547 and 0.622 respectively, which are very low.

The Residual vs fitted plot value curve as shown in figure 4.28 shows the residuals mostly concentrated around the middle line, which indicates unbiased model's prediction. The red LOESS curve indicated a small wave pattern, slightly positive for mid-range fitted values and slightly negative for higher values at the end. This is because the model is trying capture minor non-linearity. The spread of the residual points shows that the variance is very low and there are no outliers. Overall, the GAM model is fitting well with very good performance in handling the non-linearity.

4.14 LASSO Regression Model

The LASSO (Least Absolute Shrinkage and Selection Operator) regression model was built, which penalizes complexity by assigning zero value to the less important variables. The result and coefficient path obtained after running the model are presented in figure 4.29.

```
x1_Empathy = 0.6048
x2_Automation = 0.1928
x3_Vision = 0.0000
x4_SustEthics = -0.1115
x5_ReqSource = 0.0000
x6_Growth = 0.1892
x7_Human = 0.0701
x8_Scope = -0.0000
x9_Governance = 0.0864
x10_Strategy = 0.0094
x11_IssueRes = 0.0000
x12_Retrospective = 0.1633
x13_Impediments = -0.0389
x14_Integration = 0.0000
```

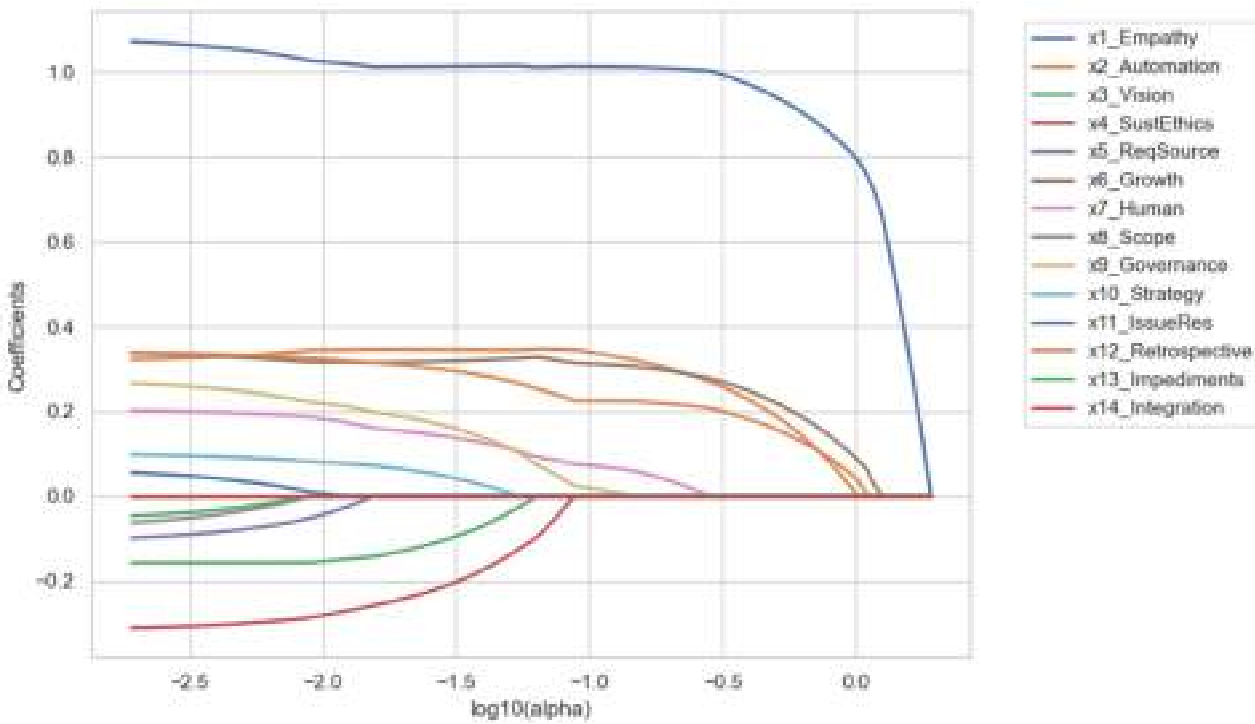


Figure 4.29 LASSO Regression – Results and Coefficient Path

The most influential independent variables are x1_Empathy (β = 0.6048), x2_Automation (β = 0.1928), x4_SustEthics (β = -0.1115), and x6_Growth (β = 0.1892). The variables with moderate impact on project success are x7_Human (β = 0.0701), x9_Governance (β = 0.0864), x10_Strategy (β = 0.0094), x12_Retrospective (β = 0.1633), and x13_Impediments (β = -0.0389). Five variables such as x3_Vision, x5_ReqSource, x8_Scope, x11_IssueRes, and x14_Integration were dropped based on zero coefficient value ($\beta = 0$). These finding are in alignment with findings from earlier models.

4.15 Robust Linear Regression Model

The robust linear regression smooths out the effect of data points those do not follow the main trend by reducing the influence of outliers and handling non-constant variance (heteroscedasticity) in a better manner. We used this technique on the dataset and derived the results as shown in figure 4.30.

We have used the Iteratively Reweighted Least Squares (IRLS) method along with HuberT norm, which is suitable for low to moderate outliers. The Scale Estimation method used is MAD (Median Absolute Deviation) to estimate the residual variability. For robust estimation, 23 number of iterations were needed so as to refine iteratively to converge.

The dominating variables are x_1_Empathy (β = 0.6445 and $p<0.001$), x_2_Automation (β = 0.2154 and $p < 0.001$), x_4_Sustainability & Ethics (β = -0.1839 and $p = 0.001$), x_6_Growth (β = 0.2636 and $p < 0.001$). In this way, this robust model verifies the results from earlier OLS output of Multiple Linear regression.

```
                    Robust linear Model Regression Results
==============================================================================
Dep. Variable:      Y_Project_Success   No. Observations:                  302
Model:                            RLM   Df Residuals:                      287
Method:                          IRLS   Df Model:                           14
Norm:                          HuberT
Scale Est.:                       mad
Cov Type:                          H1
Date:                Thu, 02 Oct 2025
Time:                        16:59:20
No. Iterations:                    23
=====================================================================================
                        coef    std err          z      P>|z|      [0.025      0.975]
-------------------------------------------------------------------------------------
const                -4.2137      0.152    -27.658      0.000      -4.512      -3.915
x1_Empathy            0.6445      0.076      8.517      0.000       0.496       0.793
x2_Automation         0.2154      0.057      3.793      0.000       0.104       0.327
x3_Vision             0.0218      0.081      0.270      0.787      -0.136       0.180
x4_SustEthics        -0.1839      0.054     -3.420      0.001      -0.289      -0.079
x5_ReqSource         -0.0642      0.070     -0.920      0.358      -0.201       0.073
x6_Growth             0.2636      0.072      3.638      0.000       0.122       0.406
x7_Human              0.0906      0.069      1.306      0.191      -0.045       0.227
x8_Scope             -0.0404      0.077     -0.527      0.598      -0.191       0.110
x9_Governance         0.1131      0.065      1.745      0.081      -0.014       0.240
x10_Strategy          0.0757      0.053      1.440      0.150      -0.027       0.179
x11_IssueRes          0.0237      0.071      0.336      0.737      -0.115       0.162
x12_Retrospective     0.1192      0.066      1.803      0.071      -0.010       0.249
x13_Impediments      -0.0655      0.056     -1.166      0.244      -0.176       0.045
x14_Integration      -0.0365      0.077     -0.473      0.636      -0.188       0.115
=====================================================================================
```

Figure 4.30 Robust Linear Regression (RLR) output

4.16 Multivariate Analysis of Variance (MANOVA) Model

The Multivariate Analysis of Variance (MANOVA) technique is used to analyze the impact of independent variables (x_1 to x_{14}) on the set of dependent variables (y_1 to y_5), rather than one Y variable or each variable y_i in isolation. The analysis was conducted using StatsModel API and libraries such as numpy and MANOVA in Python. The output is shown in figure 4.31 and 4.32.

Multivariate linear model

x0	Value	Num DF	Den DF	F Value	Pr > F
Wilks' lambda	0.0670	5.0000	283.0000	787.9233	0.0000
Pillai's trace	0.9330	5.0000	283.0000	787.9233	0.0000
Hotelling-Lawley trace	13.9209	5.0000	283.0000	787.9233	0.0000
Roy's greatest root	13.9209	5.0000	283.0000	787.9233	0.0000

x1	Value	Num DF	Den DF	F Value	Pr > F
Wilks' lambda	0.7945	5.0000	283.0000	14.6387	0.0000
Pillai's trace	0.2055	5.0000	283.0000	14.6387	0.0000
Hotelling-Lawley trace	0.2586	5.0000	283.0000	14.6387	0.0000
Roy's greatest root	0.2586	5.0000	283.0000	14.6387	0.0000

x2	Value	Num DF	Den DF	F Value	Pr > F
Wilks' lambda	0.9131	5.0000	283.0000	5.3044	0.0001
Pillai's trace	0.0869	5.0000	283.0000	5.3044	0.0001
Hotelling-Lawley trace	0.0951	5.0000	283.0000	5.3044	0.0001
Roy's greatest root	0.0951	5.0000	283.0000	5.3044	0.0001

x3	Value	Num DF	Den DF	F Value	Pr > F
Wilks' lambda	0.9787	5.0000	283.0000	1.2294	0.2954
Pillai's trace	0.0213	5.0000	283.0000	1.2294	0.2954
Hotelling-Lawley trace	0.0217	5.0000	283.0000	1.2294	0.2954
Roy's greatest root	0.0217	5.0000	283.0000	1.2294	0.2954

x4	Value	Num DF	Den DF	F Value	Pr > F
Wilks' lambda	0.9065	5.0000	283.0000	5.8353	0.0000
Pillai's trace	0.0935	5.0000	283.0000	5.8353	0.0000
Hotelling-Lawley trace	0.1031	5.0000	283.0000	5.8353	0.0000
Roy's greatest root	0.1031	5.0000	283.0000	5.8353	0.0000

x5	Value	Num DF	Den DF	F Value	Pr > F
Wilks' lambda	0.9360	5.0000	283.0000	3.8706	0.0021
Pillai's trace	0.0640	5.0000	283.0000	3.8706	0.0021
Hotelling-Lawley trace	0.0684	5.0000	283.0000	3.8706	0.0021
Roy's greatest root	0.0684	5.0000	283.0000	3.8706	0.0021

x6	Value	Num DF	Den DF	F Value	Pr > F
Wilks' lambda	0.8972	5.0000	283.0000	6.4870	0.0000
Pillai's trace	0.1028	5.0000	283.0000	6.4870	0.0000
Hotelling-Lawley trace	0.1146	5.0000	283.0000	6.4870	0.0000
Roy's greatest root	0.1146	5.0000	283.0000	6.4870	0.0000

x7	Value	Num DF	Den DF	F Value	Pr > F
Wilks' lambda	0.9622	5.0000	283.0000	2.2248	0.0521
Pillai's trace	0.0378	5.0000	283.0000	2.2248	0.0521
Hotelling-Lawley trace	0.0393	5.0000	283.0000	2.2248	0.0521
Roy's greatest root	0.0393	5.0000	283.0000	2.2248	0.0521

x8	Value	Num DF	Den DF	F Value	Pr > F
Wilks' lambda	0.9628	5.0000	283.0000	2.1840	0.0561
Pillai's trace	0.0372	5.0000	283.0000	2.1840	0.0561
Hotelling-Lawley trace	0.0386	5.0000	283.0000	2.1840	0.0561
Roy's greatest root	0.0386	5.0000	283.0000	2.1840	0.0561

x9	Value	Num DF	Den DF	F Value	Pr > F
Wilks' lambda	0.9718	5.0000	283.0000	1.6410	0.1492
Pillai's trace	0.0282	5.0000	283.0000	1.6410	0.1492
Hotelling-Lawley trace	0.0290	5.0000	283.0000	1.6410	0.1492
Roy's greatest root	0.0290	5.0000	283.0000	1.6410	0.1492

x10	Value	Num DF	Den DF	F Value	Pr > F
Wilks' lambda	0.9579	5.0000	283.0000	2.4853	0.0318
Pillai's trace	0.0421	5.0000	283.0000	2.4853	0.0318
Hotelling-Lawley trace	0.0439	5.0000	283.0000	2.4853	0.0318
Roy's greatest root	0.0439	5.0000	283.0000	2.4853	0.0318

x11	Value	Num DF	Den DF	F Value	Pr > F
Wilks' lambda	0.9957	5.0000	283.0000	0.2419	0.9436
Pillai's trace	0.0043	5.0000	283.0000	0.2419	0.9436
Hotelling-Lawley trace	0.0043	5.0000	283.0000	0.2419	0.9436
Roy's greatest root	0.0043	5.0000	283.0000	0.2419	0.9436

x12	Value	Num DF	Den DF	F Value	Pr > F
Wilks' lambda	0.8442	5.0000	283.0000	10.4455	0.0000
Pillai's trace	0.1558	5.0000	283.0000	10.4455	0.0000
Hotelling-Lawley trace	0.1846	5.0000	283.0000	10.4455	0.0000
Roy's greatest root	0.1846	5.0000	283.0000	10.4455	0.0000

x13	Value	Num DF	Den DF	F Value	Pr > F
Wilks' lambda	0.9460	5.0000	283.0000	3.2381	0.0075
Pillai's trace	0.0540	5.0000	283.0000	3.2381	0.0075
Hotelling-Lawley trace	0.0571	5.0000	283.0000	3.2381	0.0075
Roy's greatest root	0.0571	5.0000	283.0000	3.2381	0.0075

x14	Value	Num DF	Den DF	F Value	Pr > F
Wilks' lambda	0.9336	5.0000	283.0000	4.0252	0.0015
Pillai's trace	0.0664	5.0000	283.0000	4.0252	0.0015
Hotelling-Lawley trace	0.0711	5.0000	283.0000	4.0252	0.0015
Roy's greatest root	0.0711	5.0000	283.0000	4.0252	0.0015

Figure 4.31 Multivariate Linear Model – Tests and Results

```
Summary of all regression models:

=== Results for y1_Schedule ===
R-squared: 0.8218
Adjusted R-squared: 0.8131
F-statistic: 94.5350
Prob (F-statistic): 0.0000

Coefficients:
const: 0.4145 (p=0.0000)
x1_Empathy: 0.0481 (p=0.0000)
x2_Automation: 0.0105 (p=0.0452)
x3_Vision: 0.0013 (p=0.8604)
x4_SustEthics: -0.0166 (p=0.0000)
x5_ReqSource: -0.0053 (p=0.4141)
x6_Growth: 0.0166 (p=0.0134)
x7_Human: 0.0194 (p=0.0026)
x8_Scope: -0.0061 (p=0.3884)
x9_Governance: 0.0113 (p=0.0592)
x10_Strategy: 0.0071 (p=0.1466)
x11_IssueRes: 0.0020 (p=0.7576)
x12_Retrospective: 0.0195 (p=0.0015)
x13_Impediments: -0.0029 (p=0.5774)
x14_Integration: -0.0122 (p=0.0880)

=== Results for y2_Cost ===
R-squared: 0.7859
Adjusted R-squared: 0.7754
F-statistic: 75.2300
Prob (F-statistic): 0.0000

Coefficients:
const: 0.4607 (p=0.0000)
x1_Empathy: 0.0564 (p=0.0000)
x2_Automation: 0.0078 (p=0.1745)
x3_Vision: 0.0025 (p=0.7623)
x4_SustEthics: -0.0130 (p=0.0117)
x5_ReqSource: -0.0123 (p=0.0813)
x6_Growth: -0.0014 (p=0.8531)
x7_Human: 0.0141 (p=0.0460)
x8_Scope: -0.0040 (p=0.6035)
x9_Governance: 0.0132 (p=0.0447)
x10_Strategy: 0.0027 (p=0.6084)
x11_IssueRes: 0.0035 (p=0.6243)
x12_Retrospective: 0.0314 (p=0.0000)
x13_Impediments: -0.0128 (p=0.0277)
x14_Integration: -0.0024 (p=0.7574)

=== Results for y3_Quality ===
R-squared: 0.7975
Adjusted R-squared: 0.7876
F-statistic: 80.7261
Prob (F-statistic): 0.0000

Coefficients:
const: 0.5468 (p=0.0000)
x1_Empathy: 0.0392 (p=0.0000)
x2_Automation: 0.0168 (p=0.0002)
x3_Vision: -0.0047 (p=0.4452)
x4_SustEthics: -0.0086 (p=0.0380)
x5_ReqSource: -0.0029 (p=0.5865)
x6_Growth: 0.0129 (p=0.0203)
x7_Human: 0.0082 (p=0.1222)
x8_Scope: -0.0069 (p=0.2367)
x9_Governance: 0.0128 (p=0.0100)
x10_Strategy: 0.0084 (p=0.0366)
x11_IssueRes: 0.0018 (p=0.7416)
x12_Retrospective: 0.0026 (p=0.6009)
x13_Impediments: -0.0037 (p=0.3833)
x14_Integration: -0.0039 (p=0.5001)

=== Results for y4_Scope ===
R-squared: 0.7715
Adjusted R-squared: 0.7604
F-statistic: 69.2176
Prob (F-statistic): 0.0000

Coefficients:
const: 0.5763 (p=0.0000)
x1_Empathy: 0.0354 (p=0.0000)
x2_Automation: 0.0064 (p=0.1717)
x3_Vision: -0.0008 (p=0.9082)
x4_SustEthics: -0.0164 (p=0.0002)
x5_ReqSource: -0.0123 (p=0.0328)
x6_Growth: 0.0283 (p=0.0000)
x7_Human: 0.0028 (p=0.6400)
x8_Scope: 0.0087 (p=0.1683)
x9_Governance: 0.0062 (p=0.2451)
x10_Strategy: -0.0005 (p=0.9106)
x11_IssueRes: 0.0032 (p=0.5769)
x12_Retrospective: 0.0071 (p=0.1927)
x13_Impediments: -0.0083 (p=0.0747)
x14_Integration: 0.0065 (p=0.3871)

=== Results for y5_Customer ===
R-squared: 0.8250
Adjusted R-squared: 0.8164
F-statistic: 96.4263
Prob (F-statistic): 0.0000

Coefficients:
const: 0.0940 (p=0.0000)
x1_Empathy: 0.0701 (p=0.0000)
x2_Automation: 0.0276 (p=0.0004)
x3_Vision: -0.0170 (p=0.1242)
x4_SustEthics: -0.0235 (p=0.0001)
x5_ReqSource: 0.0109 (p=0.2565)
x6_Growth: 0.0149 (p=0.1344)
x7_Human: 0.0117 (p=0.2201)
x8_Scope: -0.0174 (p=0.0992)
x9_Governance: 0.0229 (p=0.0103)
x10_Strategy: 0.0101 (p=0.1622)
x11_IssueRes: 0.0084 (p=0.3833)
x12_Retrospective: 0.0156 (p=0.0065)
x13_Impediments: -0.0133 (p=0.0838)
x14_Integration: 0.0210 (p=0.0472)
```

Figure 4.32 MANOVA Regression Model Results

Various multi-variate tests such as Wilks, Pillai, Hotelling, and Roy tests, as shown in figure 4.31, were performed on the dataset to check whether each predictor (x_1 to x_{14}) significantly explains variation across all dependent variables together (y_1 to y_5).

Based on analyzing that, whether the p value is less than 0.05, we found that:

Significant independent variable ($p < 0.05$): x1_Empathy, x2_Automation, x4_Sustainability & Ethics, x5_ReqSource, x6_Growth, x10_Strategy, x12_Retrospective, x13_Impediments, and x14_Integration. Whereas, the non-significant independent variables ($p > 0.05$): x3_Vision, x7_Human, x8_Scope, x9_Governance, x11_IssueRes. These variables don't explain much variance jointly across project outcomes.

The summary of all the regression models is shown in figure 4.32. The R^2 values are found for Schedule (0.8218), Cost (0.7859), Quality (0.7876), Scope (0.7715), Customer satisfaction (0.8250). So, it mean, the model is very strong and explains approximately 77 to 83% of the variance in project success dimensions.

Also, the p values for each of the independent variables (x_1 to x_{14}) on every dependent variable (y_1 to y_5) is plotted as shown in figure 4.33. We have also categorized and marked the p values less than 0.001 (Most significant), 0.01 (Significant) and 0.05 (borderline significant). It is observed that, the Empathy (x1) is the most significant factor. Improvement in this one variable will positively impact all five success parameters for success.

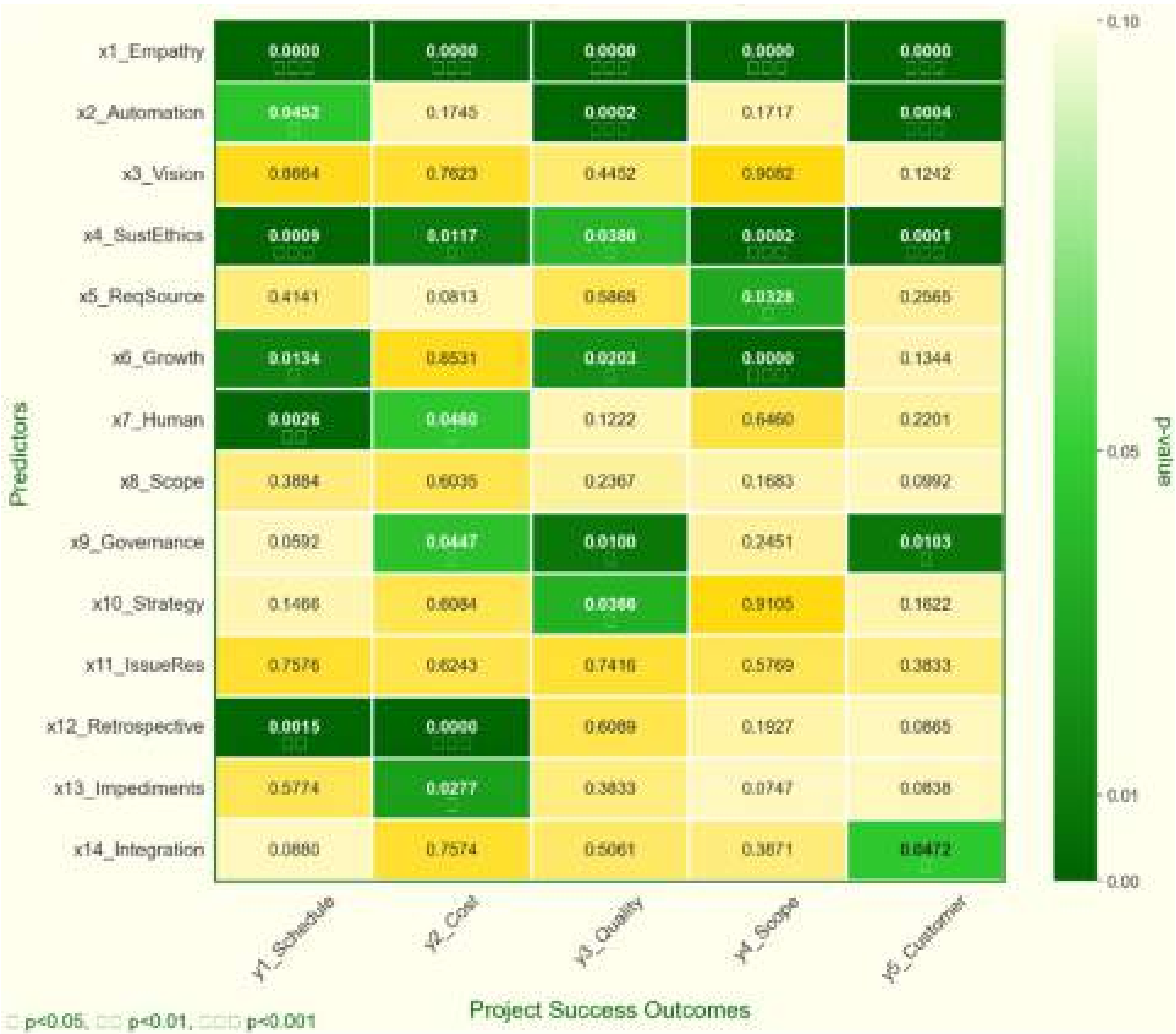


Figure 4.33 MANOVA – p Value heatmap between x_i and y_i

The coefficients for these combinations is also shown as a heatmap in figure 4.34. The coefficients for Empathy are high value and positive.

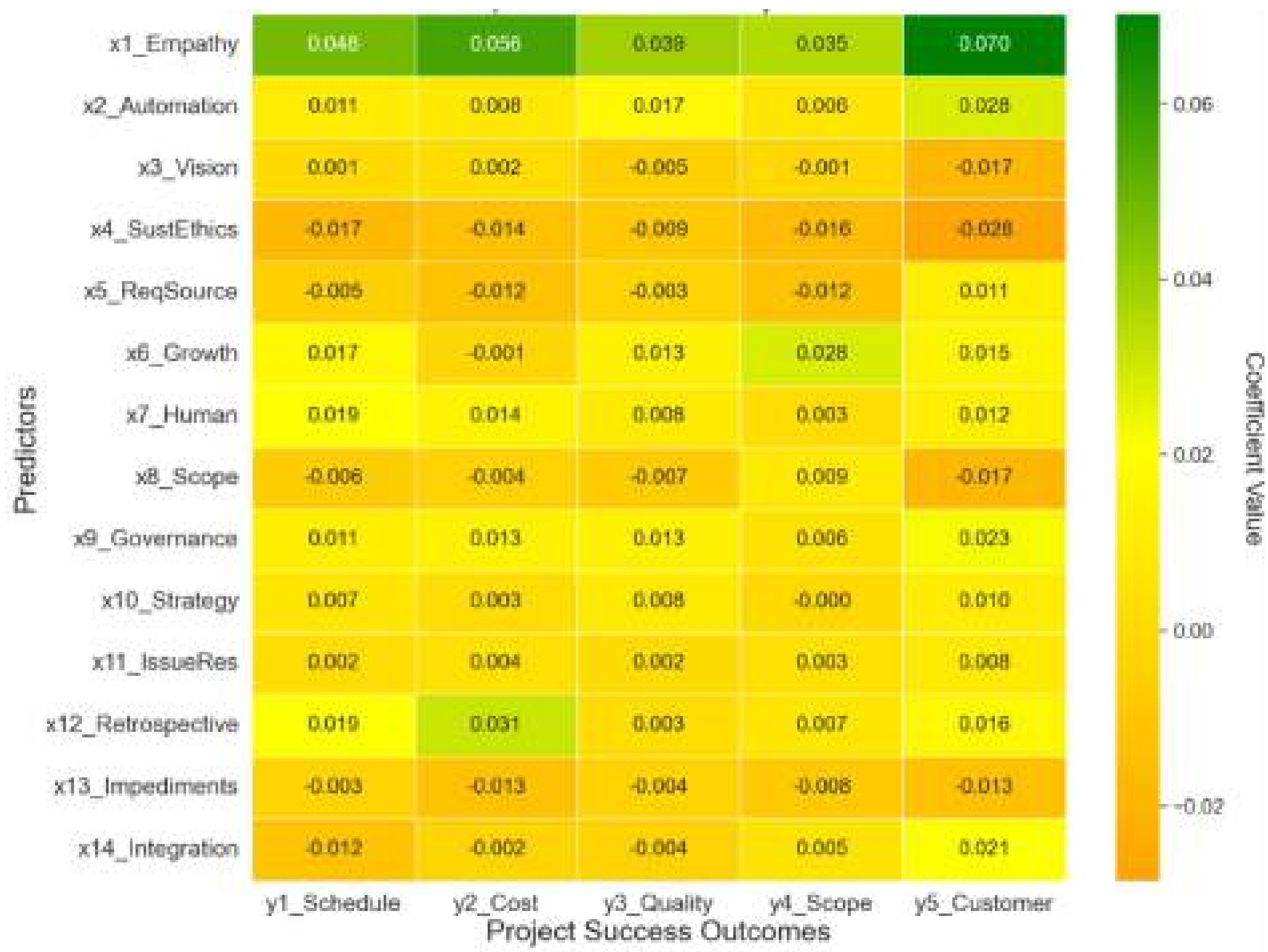


Figure 4.34 MANOVA – Coefficient heatmap between x_i and y_i

The variable Automation (x_2) is positive for schedule, quality, and customer satisfaction (significant in most cases). The process automation reduces delays and errors, and boost speed, efficiency and customer value. Human centered design (x_7) has positive impact on schedule and cost of the project, as it reduces the rework and increases delivery speed. Growth (x_6) has a strong positive value for schedule, quality and scope but less consistent for others, which reflects strong scope documentation as a part of growth strategy. The program governance (x_9) is significant for cost, quality, and customer satisfaction. Strong governance results into discipline and outcome assurance. The feedback integration in process from retrospective meetings (x_{12}) is significant for cost and schedule. Such continual learning loops improve efficiency and cost control. The sustainability & ethics (x_4) is unique as it is significant for all five y_i, however, it has consistently negative coefficients. This might suggest that, organizations perceive sustainability and ethical practices as resource-consuming, slowing delivery or adding costs, even if they have strategically long-term benefits. Also, the steps for sustainability and ethics are generally defined as a corporate policy on code of conduct.

4.17 One-Way ANOVA

One-Way ANOVA (Analysis of Variance) was conducted on the project dataset and found the results as shown in figure 4.35 below.

	sum_sq	df	F	PR(>F)
C(x1_Empathy)	1143.213734	5.0	274.562447	7.012107e-109
Residual	246.494937	296.0	NaN	NaN

	sum_sq	df	F	PR(>F)
C(x2_Automation)	988.335234	5.0	145.77309	1.228735e-77
Residual	401.373436	296.0	NaN	NaN

	sum_sq	df	F	PR(>F)
C(x3_Vision)	825.972957	5.0	86.738516	6.423047e-56
Residual	563.735713	296.0	NaN	NaN

	sum_sq	df	F	PR(>F)
C(x4_SustEthics)	548.390152	5.0	38.587879	1.903712e-30
Residual	841.318518	296.0	NaN	NaN

	sum_sq	df	F	PR(>F)
C(x5_ReqSource)	774.920522	5.0	74.61968	2.181988e-50
Residual	614.788148	296.0	NaN	NaN

	sum_sq	df	F	PR(>F)
C(x6_Growth)	1060.401934	5.0	190.630156	2.597673e-90
Residual	329.306736	296.0	NaN	NaN

	sum_sq	df	F	PR(>F)
C(x7_Human)	845.920505	5.0	92.091916	3.217627e-58
Residual	543.788165	296.0	NaN	NaN

	sum_sq	df	F	PR(>F)
C(x8_Scope)	597.740595	5.0	44.681401	2.814591e-34
Residual	791.968075	296.0	NaN	NaN

	sum_sq	df	F	PR(>F)
C(x9_Governance)	786.30548	5.0	77.144578	1.402146e-51
Residual	603.40319	296.0	NaN	NaN

	sum_sq	df	F	PR(>F)
C(x10_Strategy)	487.318591	5.0	31.969833	5.108325e-26
Residual	902.390079	296.0	NaN	NaN

	sum_sq	df	F	PR(>F)
C(x11_IssueRes)	522.869487	5.0	35.7089	1.478602e-28
Residual	866.839183	296.0	NaN	NaN

	sum_sq	df	F	PR(>F)
C(x12_Retrospective)	1009.408717	5.0	157.131221	4.330775e-81
Residual	380.299953	296.0	NaN	NaN

	sum_sq	df	F	PR(>F)
C(x13_Impediments)	357.355364	5.0	20.492439	1.441931e-17
Residual	1032.353306	296.0	NaN	NaN

	sum_sq	df	F	PR(>F)
C(x14_Integration)	725.886318	5.0	64.734895	1.694738e-45
Residual	663.822352	296.0	NaN	NaN

	sum_sq	df	F	PR(>F)
C(Method)	934.561713	77.0	5.97329	5.827868e-26
Residual	455.146958	224.0	NaN	NaN

Figure 4.35 One-Way ANOVA Result

The p values for all the 14 variables are less than 0.05, so all the independent variables (x_i) significantly impact project success (Y) at 95% confidence interval.

4.18 Structural Equation Modeling (SEM)

We ran the SEM (Structural Equation Modeling) model and the output is shown in figure 4.36. The output shows how the independent variables (x_1 to x_{14}) influence the latent variable of Project Success (Y), and how in turn it affects each of the observable sub-dependent variables (y_1 to y_5). Here the latent variable is Y which is being predicted by independent variables x_1 to x_{14}. The sub-dependent observable variables y_1 to y_5 are outcomes that measure the project success.

In the outputs as shown in figure 4.36, the ‘lval’ is a left-hand variable which is a dependent variable and ‘rval’ is a right-hand variable which is independent variable or predictor. The ‘op’ is an operator, for which single tilde sign (~) means regression or path from rval to lval, whereas double tilde sign (~ ~) means variance (if rval = lval) and covariance (if rvals ≠ lval). The ‘estimate’ is path coefficient for effect size from rval to lval (positive effect for positive sign and negative effect for negative sign). The ‘std. error’ is standard error for the estimate. The ‘z-value’ is the measure of significance and ‘p-value’ is the significance of estimate (significant if $p > 0.05$).

The variable with significant positive effect (p value > 0.05 and positive estimate) are: x_1_Empathy, x_2_Automation, x_6_Growth, x_7_Human, x_9_Governance, and x_{12}_Retrospective. The project success increases with increase in these variables.

	lval	op	rval	Estimate	Std. Err	z-value	p-value
0	project_success	~	x1_Empathy	0.050829	0.00623	8.159225	0.0
1	project_success	~	x2_Automation	0.014107	0.004659	3.027639	0.002465
2	project_success	~	x3_Vision	-0.002414	0.006609	-0.365226	0.714943
3	project_success	~	x4_SustEthics	-0.015204	0.004413	-3.444916	0.000571
4	project_success	~	x5_ReqSource	-0.005432	0.005725	-0.948722	0.342762
5	project_success	~	x6_Growth	0.013279	0.005942	2.23474	0.025434
6	project_success	~	x7_Human	0.013294	0.005688	2.337055	0.019436
7	project_success	~	x8_Scope	-0.006144	0.006287	-0.977129	0.328505
8	project_success	~	x9_Governance	0.013686	0.005319	2.573073	0.01008
9	project_success	~	x10_Strategy	0.006747	0.004313	1.56448	0.117705
10	project_success	~	x11_IssueRes	0.003075	0.005797	0.530437	0.595809
11	project_success	~	x12_Retrospective	0.015766	0.005424	2.906504	0.003655
12	project_success	~	x13_Impediments	-0.006915	0.004609	-1.500128	0.133581
13	project_success	~	x14_Integration	-0.003359	0.006325	-0.531093	0.595354
14	y1_Schedule	~	project_success	1.000000	-	-	-
15	y2_Cost	~	project_success	0.990568	0.020366	48.638546	0.0
16	y3_Quality	~	project_success	0.775883	0.014775	52.513981	0.0
17	y4_Scope	~	project_success	0.718068	0.023271	30.856822	0.0
18	y5_Customer	~	project_success	1.449953	0.03507	41.344517	0.0
19	project_success	~~	project_success	0.003930	0.000367	10.710644	0.0
20	y1_Schedule	~~	y1_Schedule	0.001449	0.000166	8.751359	0.0
21	y2_Cost	~~	y2_Cost	0.001996	0.000204	9.781469	0.0
22	y3_Quality	~~	y3_Quality	0.000919	0.000103	8.929166	0.0
23	y4_Scope	~~	y4_Scope	0.003753	0.000323	11.603655	0.0
24	y5_Customer	~~	y5_Customer	0.007133	0.00066	10.804495	0.0

Figure 4.36 SEM Output

The sustainability and Ethics (x_4) has p value of 0.0005, but the estimate coefficient is -0.0152, so the effect is negative, which means, as x_4 efforts increases the project success reduces. The x_{10}_Strategy is very marginal as its p values is 0.076 close to 0.05 at 95% confidence interval. Rest of the parameters are not significant and has no effect on project success Y.

For project success to outcome variables, the y_1_Schedule is estimate is 1 because it is fixed to latent variable scaling and rest of the four y_i will be compared with respect to it. All four outcomes y_2_Cost, y_3_Quality, y_4_Scope and y_5_Customer satisfaction have p value < 0.001 and positive estimate coefficient values. So, the project success has strong positive relationship with all the sub-dependent variables. Overall, the SEM model is strong and the insights derived are in alignment with other models.

4.19 Random Forest Model

The Random Forest Model is a machine learning technique to predict or classify based on the integrating results from many decision trees. Since many decision trees are considered, the strength and accuracy of the model are high for classification of independent variables based on importance with respect to project success.

We ran Random Forest model on our dataset and fetched output in form of feature (x_i) importance graph as shown in figure 4.37.

It is observed that, x_1_Empathy variable contributes highest at the rate of 66.2 % for project success Y. If we consider top factors contributing approximately 80% of impact on project success, there are three factors: x_1_Empaty (66.2%), x_{12}_Retrospecive (9.1%), and x_2_Automation (3.9%). This means that, with greater focus on empathizing and understanding customer needs and wants, the software projects have greater chance of success. Also, integrating lessons learnt from retrospective sessions back in project execution and introducing project process automation also contributing positively to project success.

Two more additional factors which contribute additional 5% success are x_9_Governnce (3.9%) and x_6_Growth (2.2%). So, focusing overall program governance, and growth and scale-up strategies right from the start of the project also positively contributes to the project success.

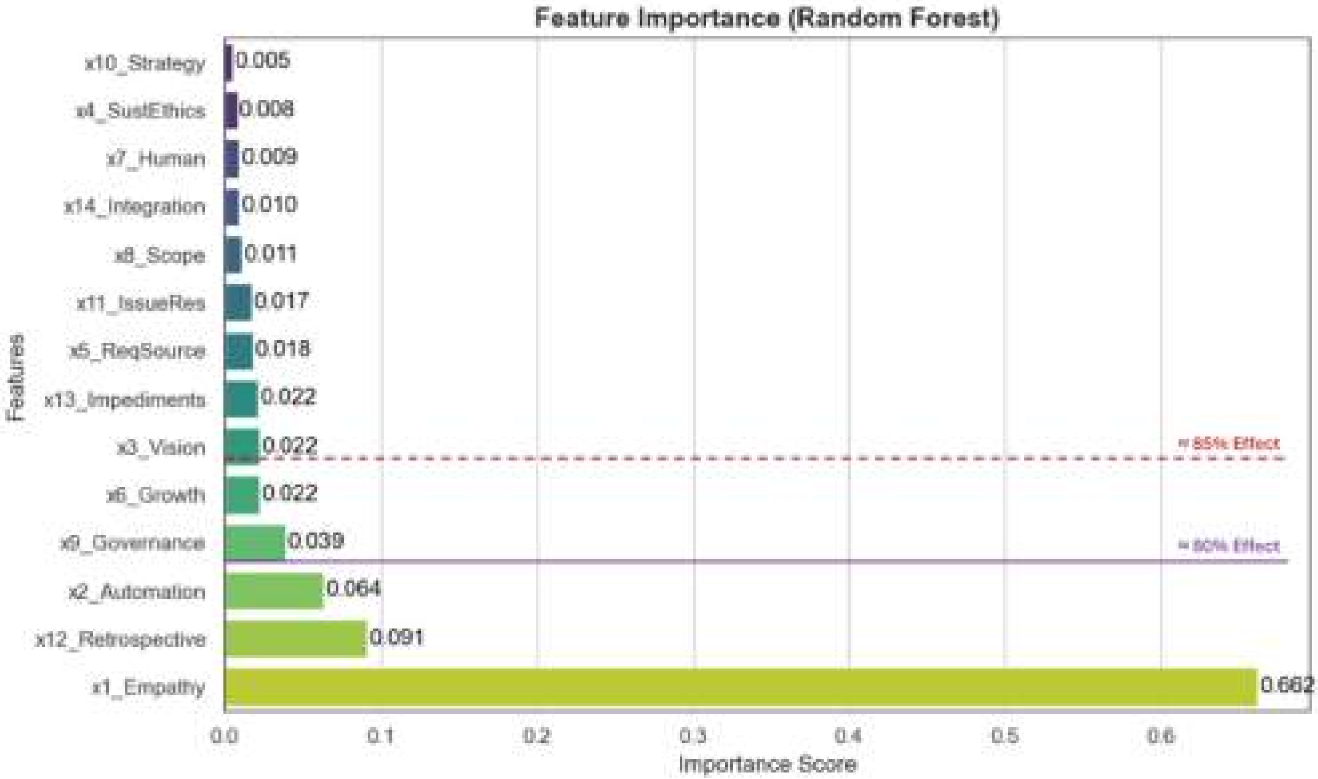


Figure 4.37 Random Forest Model – Feature Importance Graph

The RF model performance is measured via Actual vs predicted plot, and residual plot as shown in figure 4.38. The spread of the predicted value with respect actual values of test data is very much aligned and there are no major deviations.

```
Random Forest (Training) Results:
  R² Score: 0.9853
  RMSE: 0.2608
------------------------------
Random Forest (Test) Results:
  R² Score: 0.9523
  RMSE: 0.4572
------------------------------
```

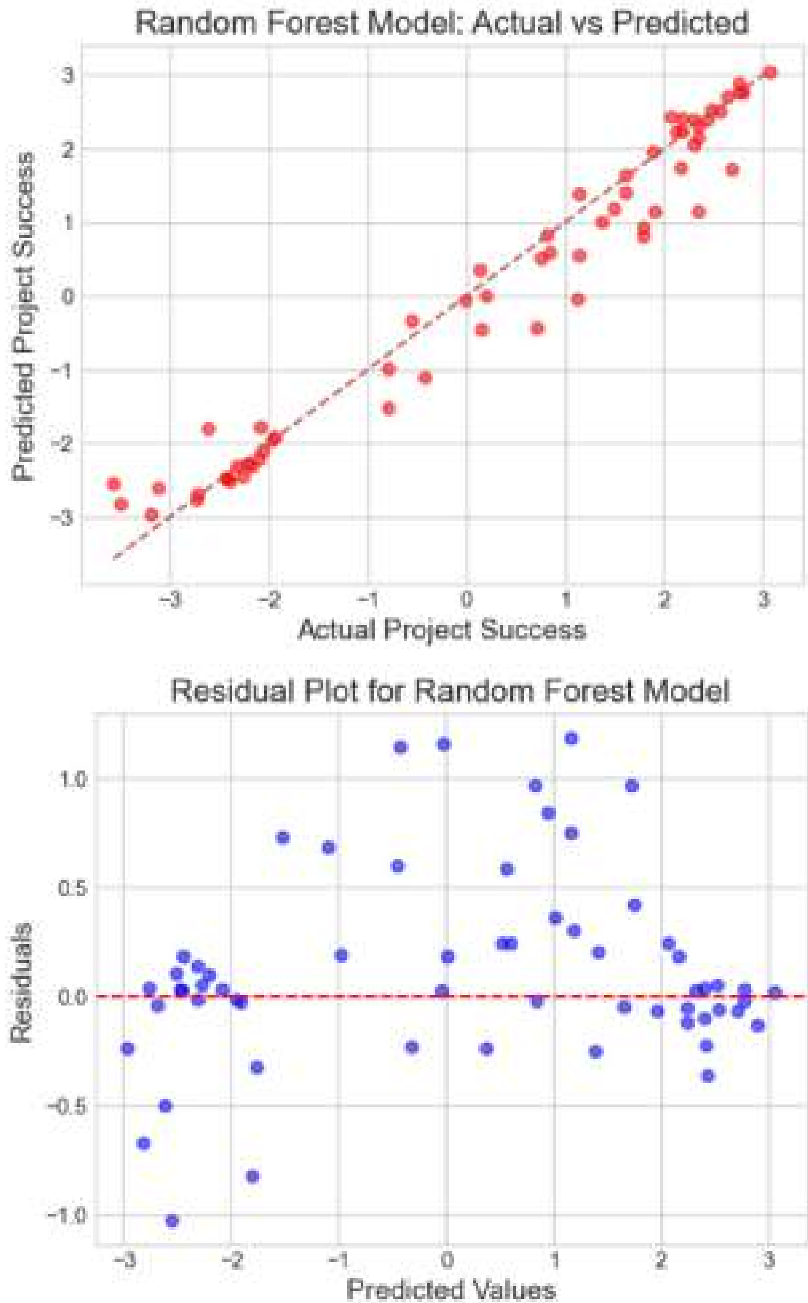


Figure 4.38 Random Forest Model – Performance

Coefficient of variance (R^2) and RMSE error scores are calculated using Python code and shown in table 4.12.

Table 4.12 Random Forest Model – Performance Analysis Metrics

Metric for Random Forest	Training	Testing
R^2 (Coefficient of Determination)	0.9853	0.9523
RMSE (Root Mean Squared Error)	0.2608	0.4572

The error in predicted test data is very low and model explains 95% of variance in the tests data. Overall, the Random Forest model built has a very good performance in predicting the project success based on input x_i variables.

4.20 XGBoost (Extreme Gradient Boosting) Model

Extreme Gradient Boosting or XGBoost model is a type of machine learning algorithm used for classification i.e. predicting categories in the data as well as regression i.e. predicting numerals. It uses gradient descent technique from mathematics to minimize errors and adjust the decision trees to improve decision at each step.

It starts with one run to get average value, finds error for the predicted output and then re-run the model to fix the error. This is done multiple times iteratively to get a more accurate and optimized model.

XGBoost analysis was conducted on the project dataset and a feature importance graph was obtained as shown in figure 4.39 by using classification algorithm.

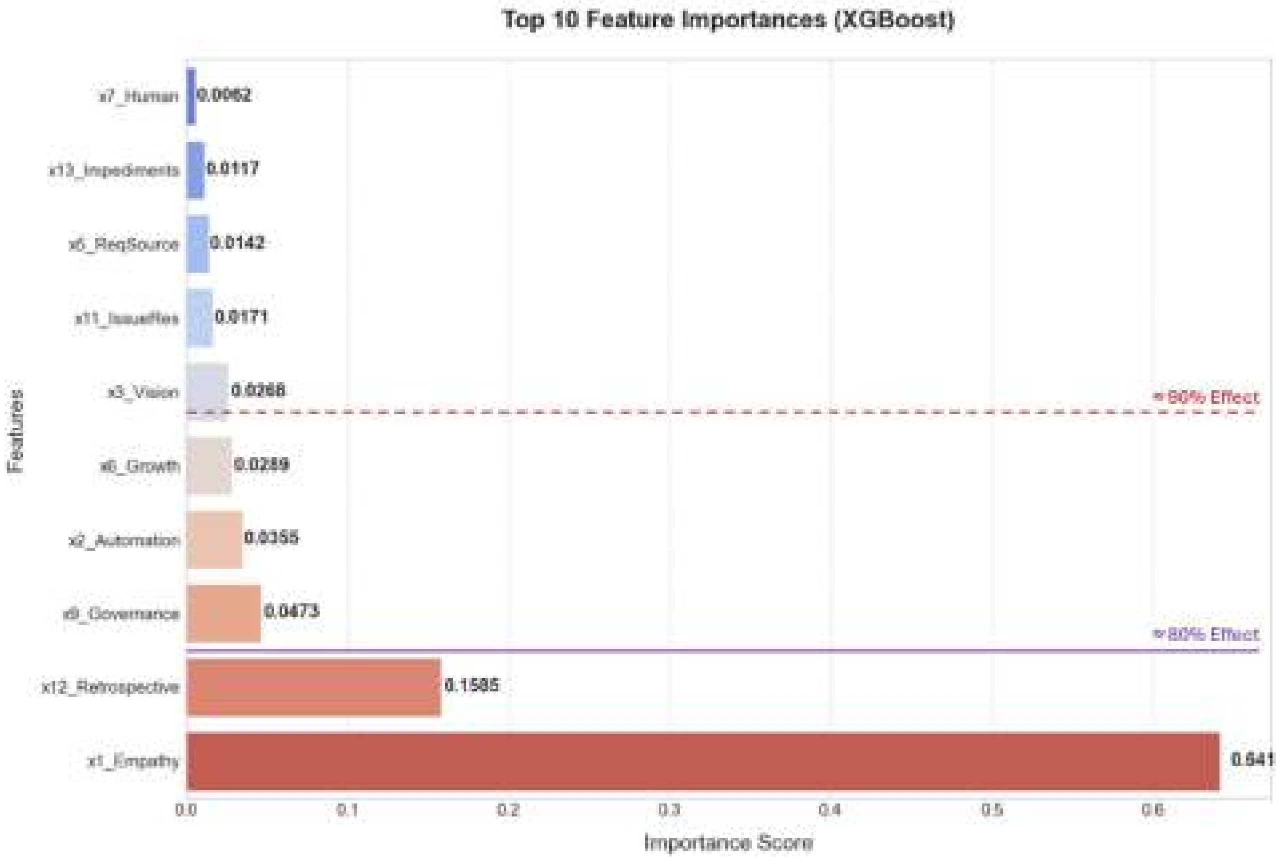


Figure 4.39 XGBoost Model – Feature Importance Graph

It is found that, the x_1_Empaty (64.15%), x_{12}_Retrospecive (15.18%) are most important factors contributing 80% of project success. x_9_Governance (4.7%), x_2_Automation (3.5%), and x_6_Growth (2.9%) are additional factors to push the project success to 91.17%. These results are aligned to Random Forest model except the fact that, XGBoost model has periodized x_9_Goveranance over x_2_Automation. Overall, this iterative process used in algorithms made XGBoost model very strong and accurate to predict the most important feature which positively impacts project success Y.

XGBoost model performance is measured by plotting the predicted data points with respect to the actual values and observing the deviation as shown in figure 4.40.

```
XGBoost (Training) Results:
  R² Score: 0.9994
  RMSE: 0.0524
------------------------------
XGBoost (Test) Results:
  R² Score: 0.9106
  RMSE: 0.6260
------------------------------
```

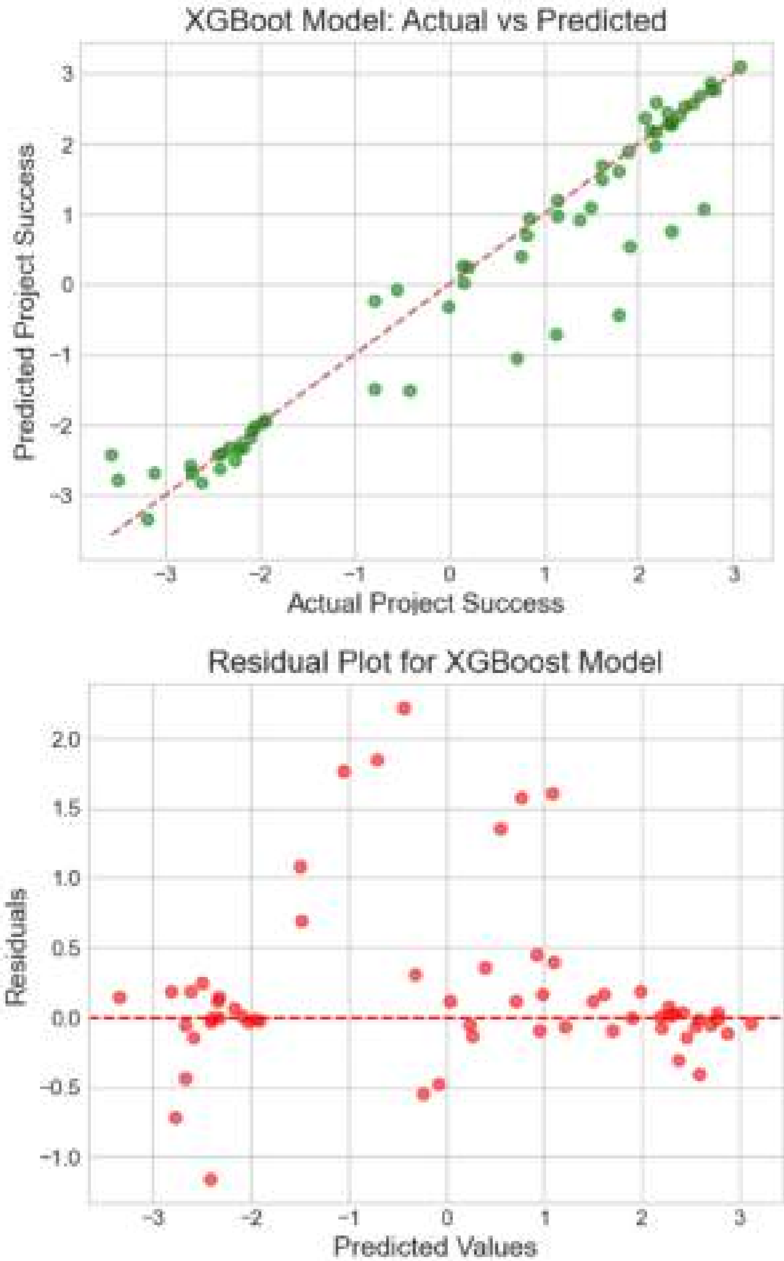


Figure 4.40 XGBoost Model – Performance

The deviation of the predicted value from actual test data is very minor, also the residual plot has a very good scatter of data points. It indicates very good model efficiency.

Table 4.13 XGBoost Model – Performance Analysis Metrics

Metric for Random Forest	Training	Testing
R^2 (Coefficient of Determination)	0.9994	0.9106
RMSE (Root Mean Squared Error)	0.0524	0.6260

The model performance metrics such as R2 and RMSE are also fairly good showing low error in prediction and better explanation capability of variability in the dataset.

4.21 Support Vector Regressor (SVR) Model

Support Vector Regressor (SVR) is a machine learning model which is used to perform regression to predict a continuous variable such as project success value. It draws a line or a plane very similar to Support Vector Machine (SVM), but to fit the datapoints while increasing the accuracy.

The line fitment differs from liner regression (which try to fit the line as close as possible to maximum data points). SVR create a tube type enclosure, which is called as tolerance margin which include variety of datapoints with certain minimized error. The points which are inside are not penalized and points outside the margin are penalized. SVR focuses on the points on the margin, which are called as support vector and these points finally define the end model. In this way, SVR creates a generalized model by creating a simple line as flat as possible by optimizing to include as many points as possible in the margin.

We split the data into train and test categories. Then learnt the scaling parameters from training data and then applied the same scaling to test data. After running the model, we found performance parameters as shown in figure 4.41

```
SVR (Training) Results:
  R² Score: 0.9275
  RMSE: 0.5797
------------------------------
SVR (Test) Results:
  R² Score: 0.9549
  RMSE: 0.4445
------------------------------
```

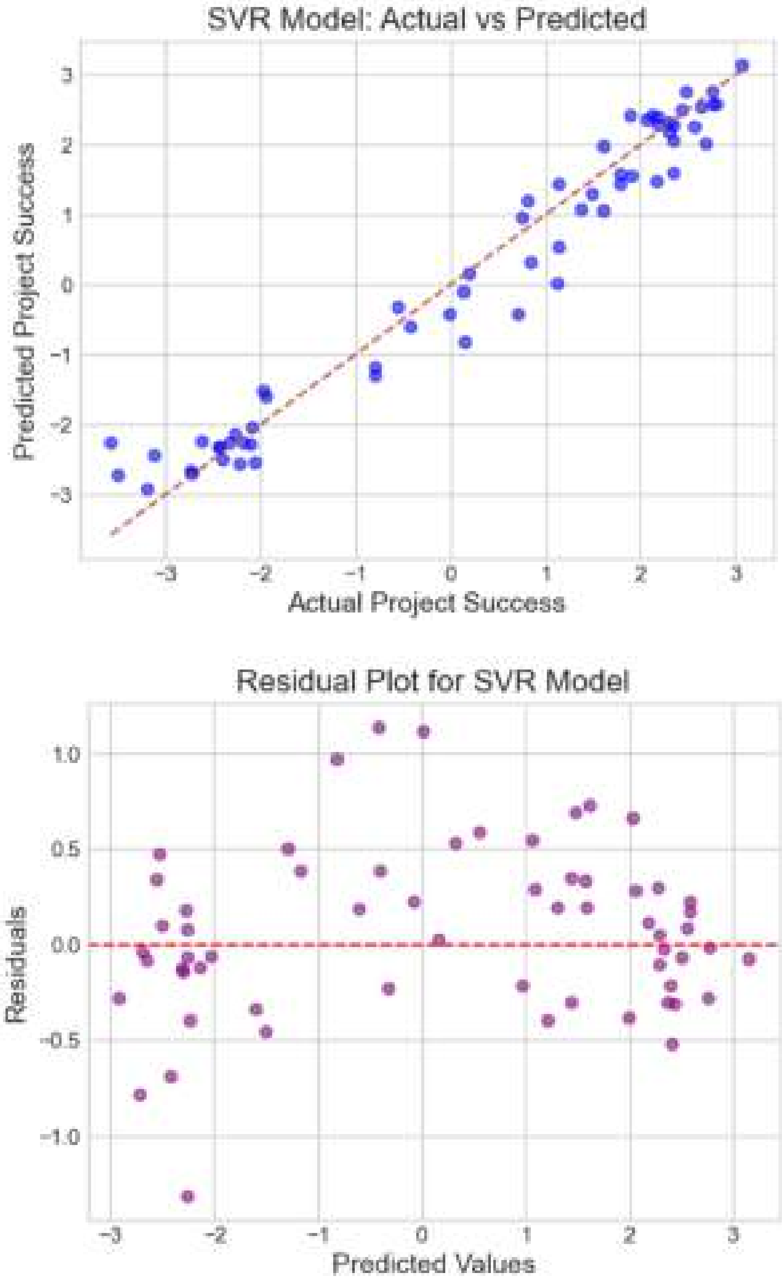


Figure 4.41 Support Vector Regressor (SVR) Model Performance

The accuracy performance of the model is shown in table 4.14. The model explains 95.49% of the variance in project success and average project performance prediction error is very low as 0.4445.

Table 4.14 Support Vector Regressor (SVR) Model Performance Metrics

Metric for SVR	Training	Testing
R^2 (Coefficient of Determination)	0.9275	0.9549
RMSE (Root Mean Squared Error)	0.5797	0.4445

Based on Actual vs Predicted values graph and residual plot as shown in figure 4.41 and performance values from table 4.14, we can conclude that, the testing performance of the model is even better in the unseen data. The model has very good fit and good project success prediction.

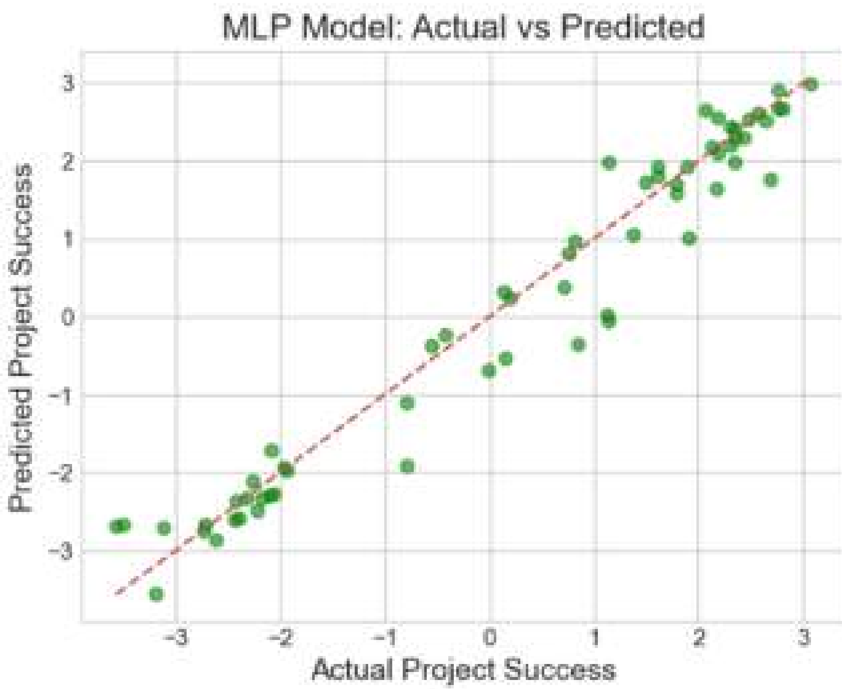
4.22 Multi-layer Perceptron (MLP) Regressor Model

Multi-layer Perceptron (MLP) Regressor is a type of ANN (Artificial Neural Network) to predict the continuous target variable. It does not classify to share important independent variable, but rather has the ability to predict the quantitative performance of the project based on input test data.

The multi-layer, here means, there are multiple layers of neurons (simply processing units). Each neuron take data input of independent variables (x_1 to x_{14}), then applies weights and biases, then process it through activation function, and finally send it to the next layer.

At each layer, the model predicts the output (Y), compare it to the actual Y, find error value, and adjust the weight accordingly. This is repeated at each layer to improve the performance and reduce the error. The final layer provide the final predicted performance value Y.

```
MLP (Training) Results:
  R² Score: 0.9969
  RMSE: 0.1196
------------------------------
MLP (Test) Results:
  R² Score: 0.9528
  RMSE: 0.4549
------------------------------
```



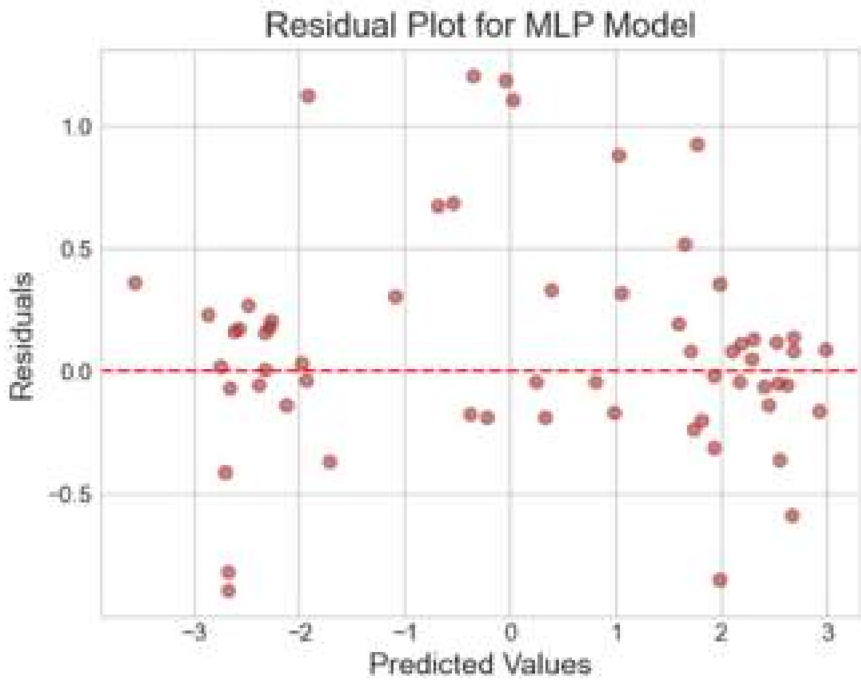


Figure 4.42 Multi-layer Perceptron (MLP) Regressor Model Performance

We ran this model on our survey dataset and observed the performance as shown in figure 4.42. The performance parameters such R2 and RMSE values are summarized in Tabe 4.15.

Table 4.15 Multi-layer Perceptron (MLP) Regressor Model Performance Metrics

Metric for MLP	**Training**	**Testing**
R^2 (Coefficient of Determination)	0.9969	0.9528
RMSE (Root Mean Squared Error)	0.1196	0.4549

MLP regressor model worked very well for our non-linear data and was able to provide very good interaction between the variables. The model's prediction accuracy is also very high and similar to SVR model.

4.23 SHAP (SHapley Additive exPlanations) Model

SVR and MLP regressor models are great at predicting the project performance Y, do not readily provide the quantitative impact of independent variable xi on project success. So, we have used the MLP regressor split (train and test data) to derive feature importance using strong SHAP algorithm.

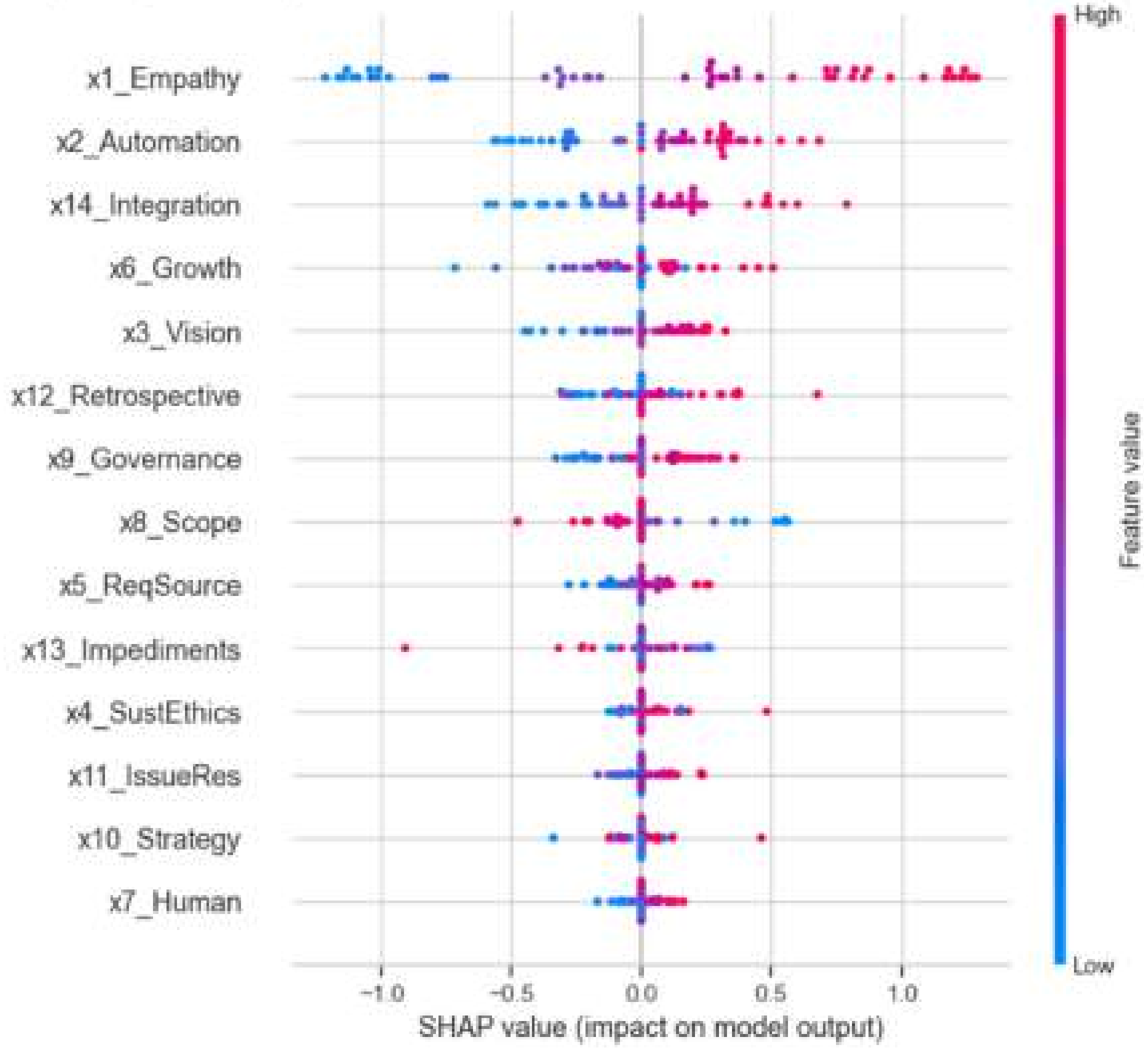


Figure 4.43 SHAP- Summary Global Feature Importance Plot

SHAP (SHapley Additive exPlanations) is not purely a model, but rather it is a model-agnostic explainability framework which helps to understand how each variable impact the predicted target variable. The SHAP concept is taken from Game Theory, where the Shapley value measures the contribution from each player to the total payoff. In our case, the players are independent variables (x_i) or features, and the total payoff can be considered as project success (Y) or predicted value. In short, it indicates the factors which increases or decreases the predicted project success score and in what magnitude.

We used, the SHAP Kernel explainer as the earlier MLP regressor is not a tree-based (needed for feature importance and categorization) algorithm. After running the SHAP algorithm, we got the output as shown in figure 4.43 and 4.44.

The global feature importance plot shows the SHAP value for each x_i which is magnitude of impact in project success Y. It shows variation in impact of each variable and its interaction with other variables. It shows the most positively impacting parameters are Empathy, Automation, Integration, Growth, Retrospective and Vision.

The Figure 4.44 shows the force plot for independent variables which indicates local push (positive) and pull (negative) on project success Y. Apart from the positively impacting feature as mentioned above, the Sustainability & Ethics variable and Impediment variable are significant but have negative effect on the project success Y.

Figure 4.44 SHAP- Force Plot

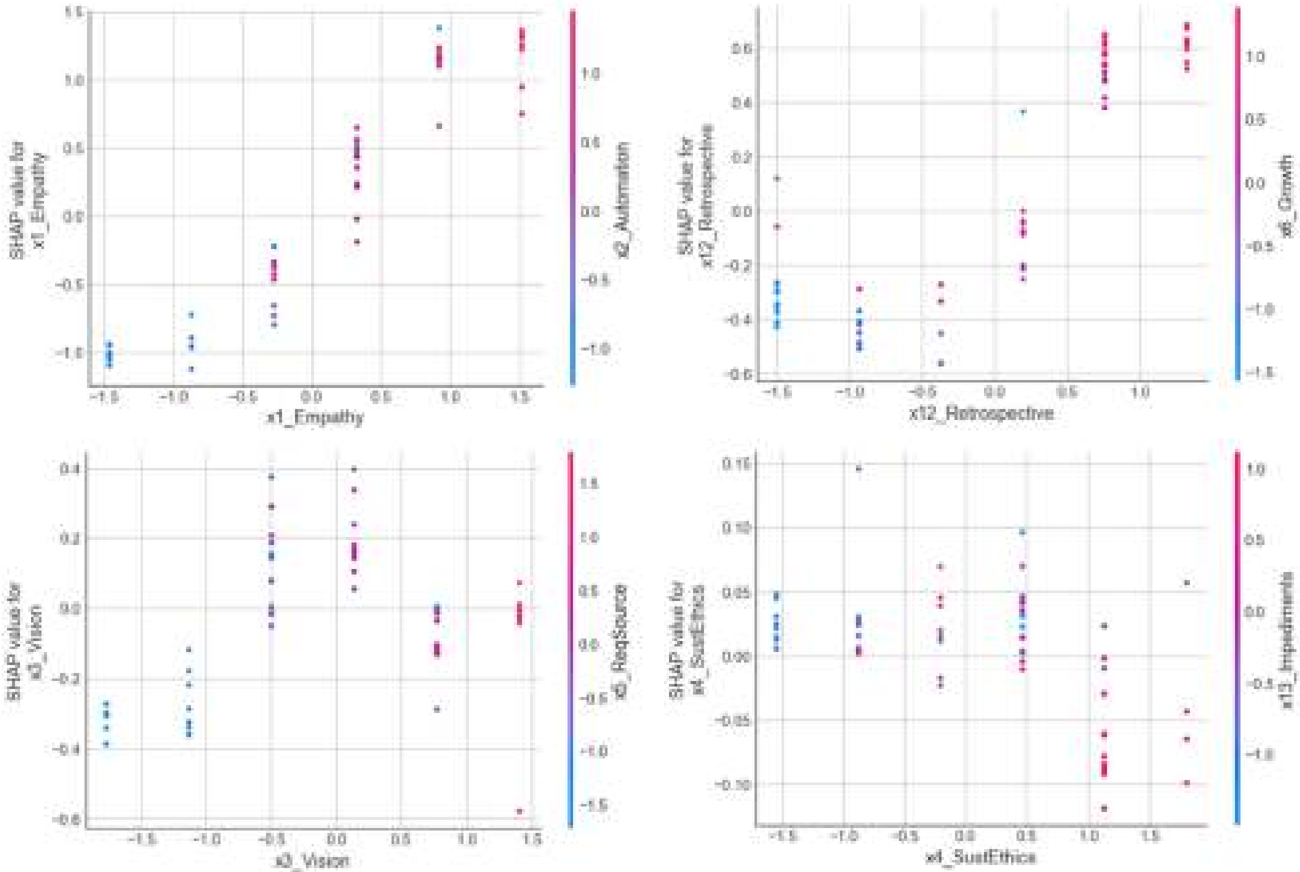


Figure 4.45 SHAP- Feature Interaction Pair Plot

The pair plots for feature-to-feature interaction are shown in figure 4.45. It shows a team with high empathy and higher level of automation performs best for project success.

The empathy and automation pair indicates the machine-human interaction strategy. For the process factor, a great growth strategy with good process to integrate lessons from retrospective in project plan performs better. Whereas on strategic factors, the higher focus on requirement collection with moderate product vision helps in predictable project success. The negative influence of Sustainability & Ethics, and impediments removal process is clearly visible form the graph.

```
SHAP (Training) Results:
  R² Score: 0.9752
  RMSE: 0.3393
------------------------------
SHAP (Test) Results:
  R² Score: 0.9192
  RMSE: 0.5951
------------------------------
```

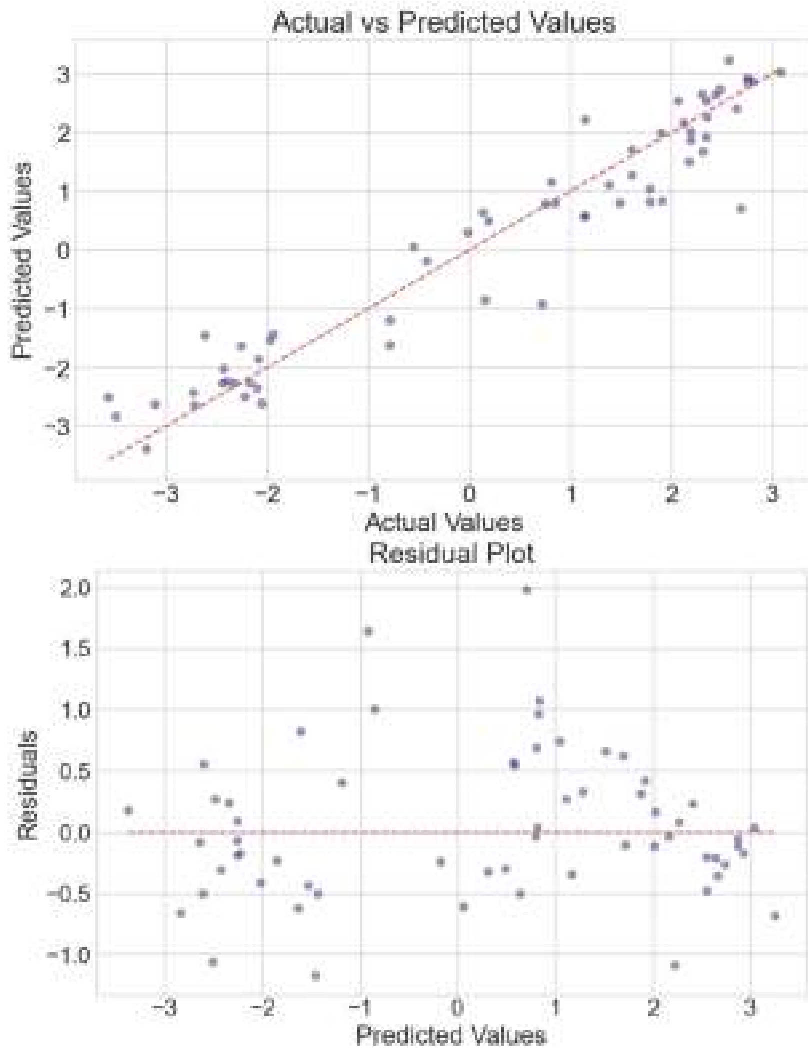


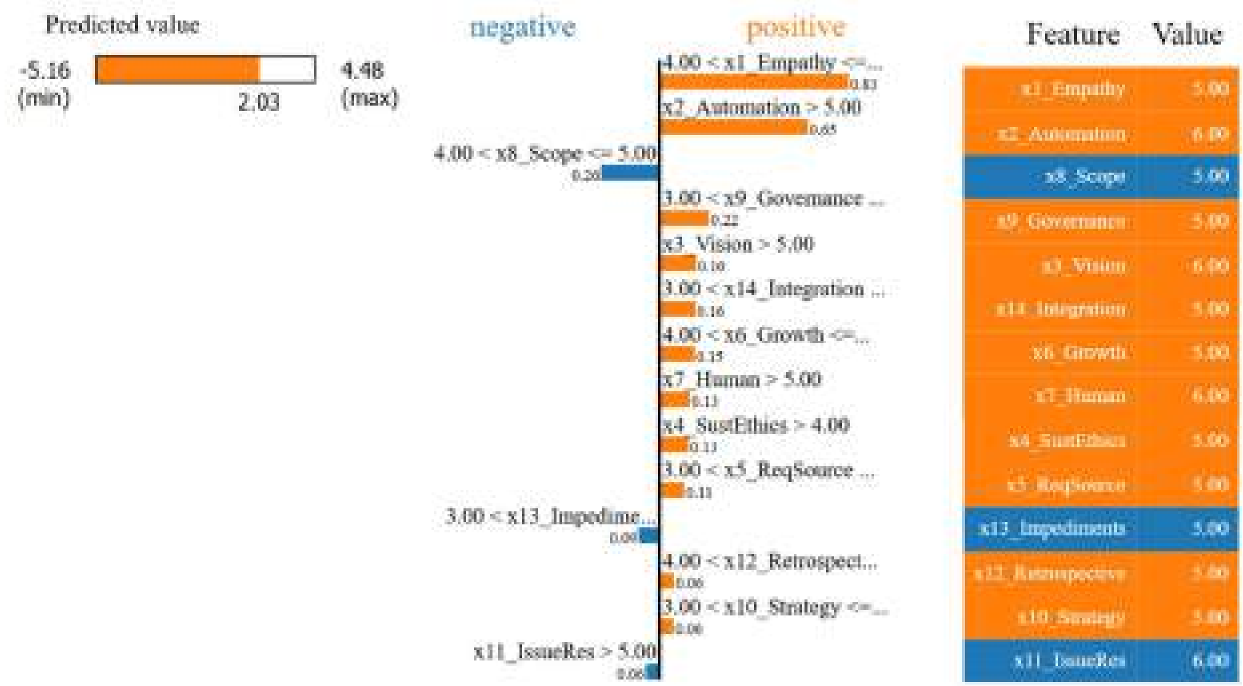
Figure 4.46 SHAP - Model Performance

The model is very strong as shown in figure 4.46 with actual and predicted values closely aligning, low error and good R-squared value.

4.24 LIME (Local Interpretable Model-agnostic Explanations) Model

LIME model-agnostics explanation technique like SHAP and is not exactly a typical model. It explains the predictions done by black-box models such as SVR, RF, ANN etc. It pick one instance of data point for explanation, create a synthetic data point, then runs our existing trained model such SVR on that synthetic data point & prediction. It fits the local interpretable model and then explain using the simple local linear model using coefficient of the governing factors.

We ran LIME algorithm and obtained the output as shown in figure 4.47. It shows the list of features with feature condition range and weight. If the weightage is positive, then it pushed the prediction up and vice versa.



[('4.00 < x1_Empathy <= 5.00', 0.83490792148418), ('x2_Automation > 5.00', 0.6542467901784009), ('4.00 < x8_Scope <= 5.00', -0.25798524133156304), ('3.00 < x9_Governance <= 5.00', 0.2189847090685001), ('x3_Vision > 5.00', 0.1611673729472719), ('3.00 < x14_Integration <= 5.00', 0.1578188840659744), ('4.00 < x6_Growth <= 5.00', 0.1546180624827727), ('x7_Human > 5.00', 0.1345579571960228), ('x4_SustEthics > 4.00', 0.1298335803003987), ('3.00 < x5_ReqSource <= 5.00', 0.1098817235121140), ('3.00 < x13_Impediments <= 5.00', -0.0918559109835971 2), ('4.00 < x12_Retrospective <= 5.00', 0.0640932770541135 4), ('3.00 < x10_Strategy <= 5.00', 0.06205785928041337), ('x11_IssueRes > 5.00', -0.0586297424178219 4)]

Figure 4.47 LIME - Model Output

The model performance in terms of R2 and RMSE value is exactly same as SHAP model, as we are not building a very new model, but explaining the existing black-box model.

Based on the feature condition range and weightage, we have analyzed and listed insights for each of the independent variable as shown in table 4.16.

Table 4.16 LIME Algorithm – Output Insights

Feature	Range	Weight	Interpretation
x1_Empathy	4 – 5	0.83	Very strong positive influence. High empathy results into project success Y.
x2_Automation	>5	0.65	Strong positive impact. More automation improves project success.
x8_Scope	4 – 5	–0.26	Negative effect. Low focus on scope slightly reduces Y.
x9_Governance	3–5	0.22	Moderate positive values help in slightly boosting project success.
x3_Vision	>5	0.16	
x14_Integration	3–5	0.16	
x6_Growth	4–5	0.15	
x7_Human	>5	0.13	
x4_SustEthics (Sustainability & Ethics)	>4	0.13	
x5_ReqSource (Requirements Sourcing)	3–5	0.11	
x13_Impediments	3–5	–0.09	Negative weightage. Some impediments reduce success, which is very practical.
x12_Retrospective	4–5	0.06	Provide small but positive contribution to success.
x10_Strategy	3–5	0.06	
x11_IssueRes (Issue Resolution)	>5	–0.06	Minor negative, due to absence of formal issue resolution process.

Overall, the LIME algorithm helps to explain the predictions with level of influence of each of the variable on target variable. The significant variable and their magnitude are quite aligned to the previous models.

4.25 Decision Tree (Supervised Learning ML Model)

Decision Tree is a supervised machine learning model which helps in predicting Y for given set of x_i by splitting the data based on conditions. We built and ran the decision tree model and obtained the output tree model as shown in figure 4.48.

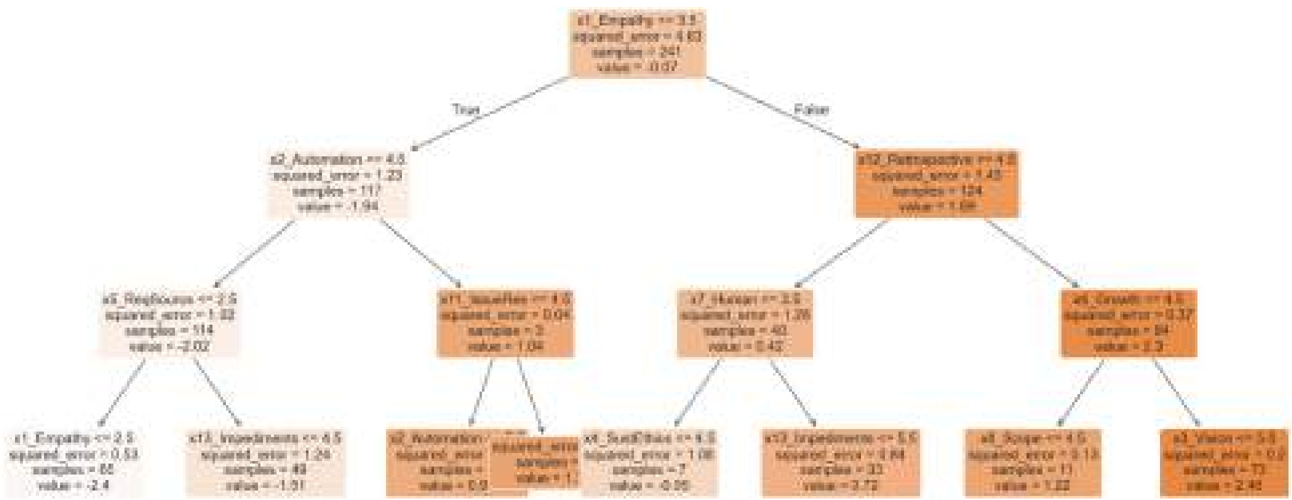


Figure 4.48 Decision Tree Model Output

The root node is x_1_Empathy with threshold value <= 3.5 and average predicted value as -0.07. It is further split into left branch of x_2_Automation based on condition x_1_Empathy <= 3.5 (True condition) and left branch x_{12}_Retrospective based on condition x_1_Empathy > 3.5 (False condition). And similarly, the subsequent nodes are split into branches based on decision rules.

We analyzed the decision tree for decision rules, threshold at each node and obtained insights as shown in table 4.17 for specific impact on project success along with line of reasoning.

Table 4.17 Decision Tree – Output Insights

Variable (x_i)	**Analysis/ Reasoning**	**Impact on Project Success Y**
x1_Empathy	Root Node split. Impact almost all predictions	Very High
x12_Retrospective	Early split in high-empathy branch	Medium-High
x2_Automation	Early split in low-empathy branch	Medium-High
x6_Growth	Split in high-empathy branch	Medium-High
x3_Vision	Leaf node in high-empathy branch	Medium-High
x5_ReqSource	Split in low-empathy branch	Medium
x13_Impediments	Appears in both branches	Medium
x7_Human	Mid-level split	Medium
x11_IssueRes	Very small sample size of 3	Low
x8_Scope	Leaf node in growth branch	Low

From decision tress, it is concluded that, the most significant variables are Empathy, Automation, Retrospective, Growth and Vision. Moderate drivers are Requirement source, Human centered design, and Impediments process. Rest of the factors do not impact significantly the project success.

4.26 Rule Mining (Unsupervised Learning ML Model)

The Rule Mining is a type of unsupervised machine learning model, which tries to find interesting pattern or relationships in the dataset without pre-defined labels or rules. It finds association such if- then, for example, if A (Antecedent) → then B (Consequent).

After running the rule mining algorithm on our dataset using Python code, we got the output as shown in figure 4.49.

Number of frequent itemsets found: 252

	antecedents	consequents	support	confidence	lift
73	(x8_Scope_High)	(x11_IssueRes_High)	0.278146	0.884211	2.871307
72	(x11_IssueRes_High)	(x8_Scope_High)	0.278146	0.903226	2.871307
70	(x11_IssueRes_High)	(x7_Human_High)	0.245033	0.795699	2.860727
71	(x7_Human_High)	(x11_IssueRes_High)	0.245033	0.880952	2.860727
230	(x7_Human_High)	(x1_Empathy_High)	0.238411	0.857143	2.724812
229	(x1_Empathy_High)	(x7_Human_High)	0.238411	0.757895	2.724812
353	(x8_Scope_High)	(x7_Human_High)	0.235099	0.747368	2.686967
352	(x7_Human_High)	(x8_Scope_High)	0.235099	0.845238	2.686967
286	(x7_Human_High)	(x3_Vision_High)	0.248344	0.892857	2.496693
287	(x3_Vision_High)	(x7_Human_High)	0.248344	0.694444	2.496693

Figure 4.49 Rule Mining Model Output

To interpret the result of rule mining for first output line, ‘x_8_Scope_High’ means if the scope is high (Antecedent) → then ‘x_{11}_IssueRes_High mean issue resolution variable is also high (Consequent). The support of 0.2781 indicates, fraction of records where this set of antecedent and consequent occur simultaneously in 27.81% of all projects. Confidence value of 0.8841 indicates, if the x_8_Scope is high, then there is 88.41 % chance that the x_{11}_IssueRes will also be high. Lift of 2.87 indicates that the x_{11}_IssueRes has 2.87 times occurrence if x_8_Scope_High is true than just by chance. The lift values are positive for all cases, so it means there is positive association.

In summary, x_8_Scope, x_{11}_IssueRes, x_7_Human, x_1_Empathy, and x_3_Vision are the most significant factors impacting the project success Y positively.

4.27 Summary and Conclusion of All Statistical Models

We have built 15 models with variety from statistical models, ANOVA, Regression models to advance Machine Learning models etc. The models were fundamentally built to serve 2 purposes – To predict the performance of the project for given input parameters, and to understand the most significant independent parameters which contributes to project success (Y).

Table 4.18 Summary of model performance parameters

		Models							
#	Model Assessment parameter	R^2 Train	RMSE Train	R^2 Test	RMSE Test	Residual vs Fitted Plot	Predicted vs Actual Y Plot	F Stat	VIF
1	Multiple Linear Regression (MLR)	0.845	0.847	0.91	0.629	Potential Non-Linearity observed	-	124.6	Very low to moderate colinearity observed
2	Generalized Additive Model (GAM)	0.935	0.547	0.912	0.622	Good fit, Low variance, No Outliers	-	-	-
3	MANOVA	-	-	0.7715 to 0.8250	-	-	-	-	-
4	Random Forest Model	0.9853	0.2608	0.9523	0.4572	Good fit, Low variance, No Outliers.	Results well aligned.	-	-
5	XGBoost Model	0.9994	0.0524	0.9106	0.626	Good fit, Low variance, No Outliers.	Results well aligned, except few minor exceptions.	-	-
6	Support Vector Regressor (SVR)	0.9275	0.5797	0.9549	0.4445	Good fit, Low variance, No Outliers.	Results overall aligned.	-	-
7	Multi-layer Perceptron (MLP) Regressor	0.9969	0.1196	0.9528	0.4549	Good fit, Low variance, No Outliers.	Results overall aligned.	-	-
8	SHAP (SHapley Additive exPlanations) Model	0.9752	0.3393	0.9192	0.5951	Good fit, Low variance, No Outliers.	Results overall aligned.	-	-

To access the prediction quality of the models, we have calculated R^2 and RMSE values for train and test data. We have also plotted Residual vs Fitted Plot and Predicted vs Actual Y Plot. We also studied F-Stat and VIF (Variance Inflation Factor) for selected model. The results are summarized in table 4.18.

All the models have very low error value (≤ 0.62) and very high R^2 values (≥ 0.91). Based on highest R^2 values and least RMSE values, the top performing models are Support Vector Regressor (SVR), Multi-layer Perceptron (MLP) Regressor, Random Forest Model and SHAP (SHapley Additive exPlanations) Model. The Residual vs Fitted Plot has good fit for all models, with no outliers and very low variance. The Predicted vs Actual Y Plot shows good alignment of predicted results with actual project success values.

Table 4.19 Summary of Significant parameter from models

#	Model	Significant Variables (Xi)						
1	Multiple Linear Regression (MLR)	x1_Empathy	x2_Automation	x12_Retrospect	x6_Growth	x9_Governance	x7_Human	x4_SustEthics
2	Generalized Additive Model (GAM)	x1_Empathy	x7_Human	x6_Growth	x2_Automation	x3_Vision	x4_SustEthics	
3	LASSO Regression Model	x1_Empathy	x7_Human	x6_Growth	x2_Automation	x3_Vision	x4_SustEthics	
4	Robust linear Model Regression	x1_Empathy	x2_Automation	x6_Growth	x4_SustEthics			
5	MANOVA	x1_Empathy	x6_Growth	x12_Retrospect	x2_Automation	x14_Integration	x5_ReqSource	
6	One-Way ANOVA - Hypothesis Testing	All xi variables						
7	Structural Equation Modeling (SEM)	x1_Empathy	x2_Automation	x6_Growth	x7_Human	x9_Governance	x12_Retrospect	x4_SustEthics
8	Random Forest Model	x1_Empathy	x12_Retrospect	x2_Automation	x9_Governance	x6_Growth		
9	XGBoost Model	x1_Empathy	x12_Retrospect	x9_Governance	x2_Automation	x6_Growth		
10	SHAP (SHapley Additive exPlanations) Model	x1_Empathy	x2_Automation	x14_Integration	x6_Growth	x12_Retrospect	x3_Vision	
11	LIME Model	x1_Empathy	x2_Automation	x9_Governance	x3_Vision	x14_Integration	x8_Scope	
12	Decision Tree (Supervised Learning ML Model)	x1_Empathy	x12_Retrospect	x2_Automation	x6_Growth	x3_Vision		
13	Rule Mining (Unsupervised Learning ML Model)	x8_Scope	x11_IssueRes	x7_Human	x1_Empathy	x3_Vision		

The summary of significant parameters from each of the model is shown in table 4.19. The seven parameters are concluded to be significant based on statistical significance (p value) and strength of the coefficient (β).

We also checked the frequency of occurrence of each of the independent variables as shown in table 4.20

Table 4.20 Summary of Significant parameter from models

Factor	Frequency of Occurrence
x1_Empathy	**12**
x2_Automation	**11**
x6_Growth	**10**
x12_Retrospect	**7**
x4_SustEthics	**5**
x7_Human	**5**
x9_Governance	**5**
x14_Integration	3
x8_Scope	2
x3_Vision	1
x5_ReqSource	1
x11_IssueRes	1

From the summary of all the models, it can be concluded that, the most significant variables which governs project success are: x1_Empathy, x2_Automation, x6_Growth, x12_Retrospect, x7_Human, x9_Governance and x4_SustEthics.

4.28 Mathematical Model based on significant variables

From Multiple Linear Regression model, we can derive following equation for these parameters:

$$Y = \beta_0 + \sum_{i=1}^{n} \beta_i x_i + \theta$$

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \ldots \beta_{14} x_{14} + \theta$$

Where,

Y is dependent variable i.e. project success

β_0 is intercept i.e. constant term

β_i are coefficients of each of the independent variables

x_i are independent variables

θ is error value

n = 1, 2, 4, 6, 7, 9, 12

Static Project Success Equation:

Substituting values from Multiple Linear Regression Analysis result, we get the following Static Project Success Equation,

$$Y = -4.1769 + 0.643\, x_1 + 0.173\, x_2 - 0.210\, x_4 + 0.2\, x_6 + 0.144\, x_7 + 0.168\, x_9 + 0.192\, x_{12} + \theta$$

This is a linear static equation in a steady state equilibrium of the system of project at a given point of time. Here, it is assumed that, the variable change in linear fashion and the statics variable are input to the project success equation showing project performance a fixed point of time.

Dynamic Project Success Equation:

However, in practical scenario, the team actually learn every day, the maturity of systems such as automation increases every day, the learnings are integrated into organizational processes.

So, a dynamic model is needed in which variation of the project performance is considered over time period, as the project performance Y actually evolves over time. So, using ordinary differential equations (ODE), a Dynamic Project Success Equation can be defined as:

$$\frac{dY}{dt} = \sum_{i=1}^{n} \alpha_i x_i(t) - \gamma Y(t) + \vartheta(t)$$

$$\frac{dY}{dt} = \alpha_1 x_1(t) + \alpha_2 x_2(t) - \alpha_4 x_4(t) + \alpha_6 x_6(t) + \alpha_7 x_7(t) + \alpha_9 x_9(t) + \alpha_{12} x_{12}(t) - \gamma Y(t) + \vartheta(t)$$

Where,

t is time

Y(t) is cumulative project success over time t

$\frac{dY}{dt}$ is rate of project success with respect to time t

α_i is Effect Coefficient i.e. the degree of empirical effect of x_i on Y

x_i are independent variables

γ is a Diminish Constant which indicates negative factors such as team fatigue, technical debt, diminishing return, system entropy etc.

ϑ is an external uncontrollable event such as market dynamics, random noise, unforeseen event, natural calamity etc.

If we substitute variable values and solve this ordinary differential equation, there are 3 possible solutions as follows:

1. If $\frac{dY}{dt} = 0$, then the project is in state of equilibrium. This result is same as Static Project Success Equation
2. If $\frac{dY}{dt} > 0$, then the project performance is constantly improving. For example, the team is learning faster, able to close stories faster, development velocity is increasing, the cash inflow is increasing with time etc.
3. If $\frac{dY}{dt} < 0$, then the project performance is decaying or degrading over time.

Non-Linear Dynamic Project Success Equation:

The earlier equations are linear as shown below:

$$Y = \beta_0 + \sum_{i=1}^{n} \beta_i x_i + \theta$$

This is linear because the relationship between x_i and Y is constant and has additive property, there is no interaction between variable, i.e. one variable does not influence other, and there is no diminishing return.

However, practically this does not happen. There is always a non-linear interactive relationship.

So, introducing the interaction term, we get,

$$Y = \beta_0 + \sum_{i=1}^{n} \beta_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \mu_{ij} x_i x_j + \theta$$

Where,

β_i is individual direct contribution coefficient for each variable x_i

μ_{ij} is an interaction coefficient between variables x_i and x_j

$x_i x_j$ are cross terms between which, the interaction is established

For example, $\mu_{12}x_1x_2$ indicates interaction between x_1 (Empathy) and x_2 (Automation). A positive interaction indicates better UI/ UX design, ease of product use, less rework and faster delivery.

We need to show, how success is evolved dynamically along with non-linear interactions, so, adding dynamic effect to this equation, we get

$$\frac{dY}{dt} = \sum_{i=1}^{n} \alpha_i x_i(t) + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \mu_{ij} x_i(t) x_j(t) - \gamma Y(t)$$

Where,

α_i is rate of effect of x_i on Y

μ_{ij} is dynamic interaction effect i.e. amplification or dampening interaction between $x_i(t)$ and $x_j(t)$ over time.

$\gamma Y(t)$ is an external uncontrollable factor of natural decay

Over time, the non-linear growth saturates. For example, retrospective i.e. learning feedback diminishes after prolonged time, as system and team become more mature. So, the saturation of each effect is,

$$f(x_i) = \frac{x_i}{1 + k_i x_i}$$

Where,

$f(x_i)$ is effective contribution of the variable to the project success.

k_i is saturation coefficient i.e. quickness of saturation effect. $k_i > 0$

When x_i is small, $f(x_i) \approx x_i$ (Linear)

When x_i is large, $f(x_i) \approx \frac{1}{k_i}$ (Saturated)

For example, in real world project, even if focus variable such as retrospection, governance, Sustainability are increased to large extent, their contribution first increase but, at the top, it diminishes and do not add any further value.

Substituting this into the above dynamic question,

$$\frac{dY}{dt} = \sum_{i=1}^{n} \alpha_i \left(\frac{x_i(t)}{1 + k_i x_i(t)}\right) + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \mu_{ij} x_i(t) x_j(t) - \gamma Y(t)$$

This final equation indicates project success over time including effect of independent variables over time, their saturation point, effect of dynamic interaction between variables, effect of any external variable.

Overall, the models built are accurate on predicting the project success for given input governing independent variable and shares critical independent variables which impacts project success. Further the mathematical model also shows the effect of non-linearity, variable interaction and saturation.

CHAPTER V:

DISCUSSION OF RESULTS AND MODEL BUILDING

5.1 Discussion of Results, Factor Analysis and Deep-Thinking Exercise

From detailed data analysis and insights from statistical and Machine Learning models, we have observed that, there seven independent variables which significantly contribute to the software development project success. For each of the factor, we analyzed the implication and accordingly decided which type of methodology can be integrated into the project management process for software. These seven factors, their implication and concluded methodology are summarized in the table 5.1 below:

Table 5.1 Factor Analysis and Methodology Conclusion

#	Factor	Implication from survey rationale	Methodology
1	x1_Empathy	Clarity on Customer Requirements: The team deeply understands the customer's problem by empathizing on their needs before beginning of development work.	Design Thinking
2	x2_Automation	Project Process Automation: The project management/ development process is highly automated with full automation in flow, development, testing, deployment to production environment.	DevOps
3	x4_SustEthics	Sustainability & Ethics Strategy: The project plan includes clearly defined steps to address sustainability and ethical considerations. The focus is on sustainability of the business for very long time rather than just immediate success.	System Thinking/ Engineering
4	x6_Growth	GTM, Growth & Scale-up Strategy: At the start of the project, the team clearly defines the plan for Go-To-Market (GTM), Growth, and business Scale-up in the future.	System Thinking/ Engineering

#	Factor	Implication from survey rationale	Methodology
5	x7_Human	Human-Centred Design: While designing the product or service, there is a strong focus on Usability & Human aspects to make the product more user friendly.	Design Thinking
6	x9_Governance	Strength of Program Governance: There is a formal program governance process for planning, execution, and tracking, including risk and change management practices, templates, and team religiously follow the process. This methodology is applicable for any type of product development process.	Agile
7	x12_Retrospect	Lessons Learnt Feedback Loop: The team perform build-measure-learn cycle in initial phases of the project. The retrospectives and lessons learnt reviews are effective and the team effectively track actionable items and integrated back into the project.	Lean Startup

Considering the resulted methodology, a Deep-Thinking session with a whiteboard was conducted. The result of the session is shown in figure 5.1. We plotted conceptual diagrams for each of these methodologies using key concepts. Then a high-level thought was given on how to organize these techniques. A more integrated, adaptable and iterative approach was chosen.

Substantial time was spent on thinking about ways to avoid waterfalling the methodology. For example, the sequential nature of current design thinking methodology can be avoided by introducing the agility and iterative development flow in the method. A great way to achieve is to introduce lean, Human Centered Design, Early hypothesis testing by feature validation, and product vision flows between Design thinking framework and Agile methodology.

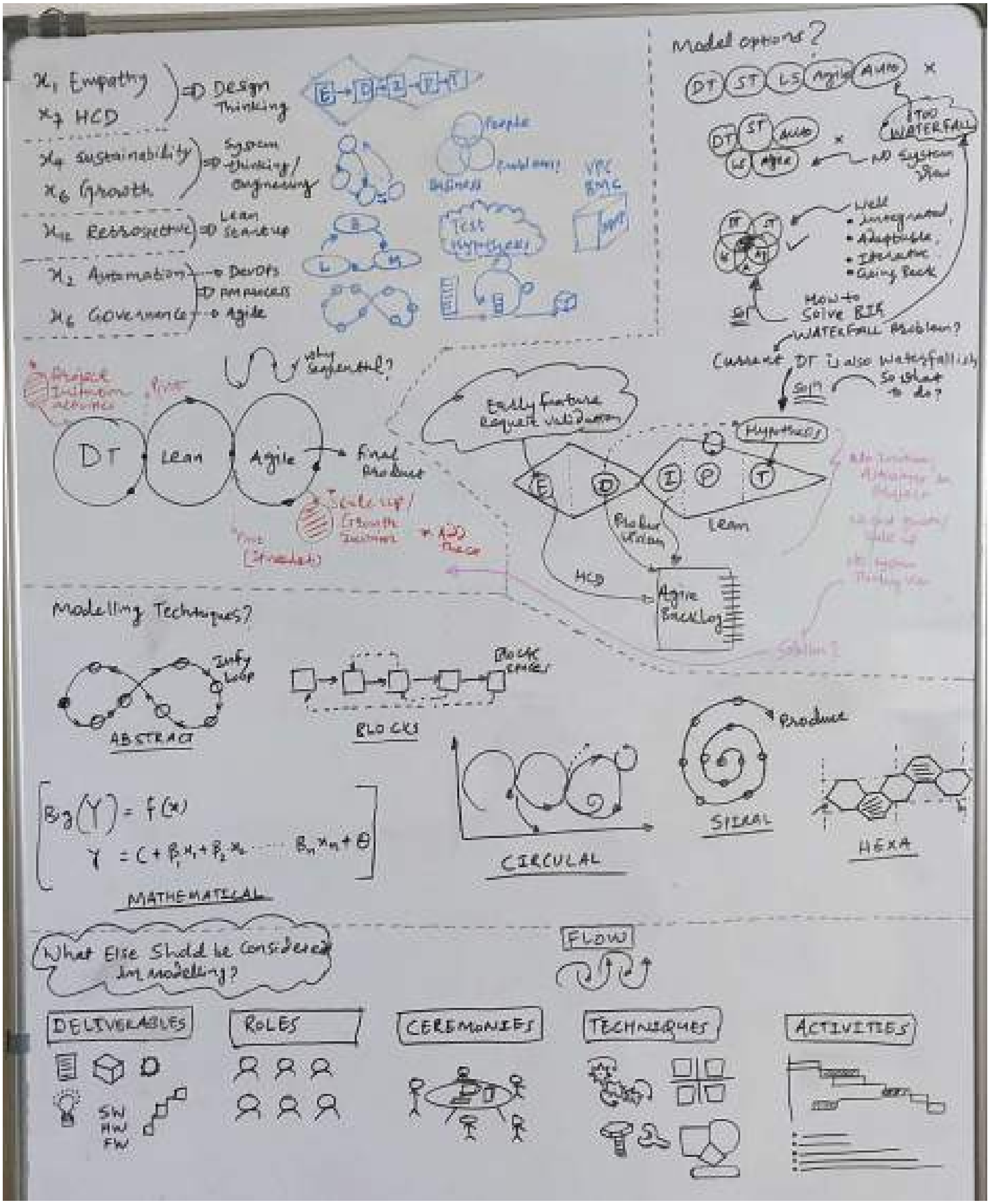


Figure 5.1 Deep-Thinking Session Whiteboard Result

During literature review, while studying of various existing models, it was observed that, the project initiation activities before start of the project and business scale-up, growth and sustainability activities post project are not properly integrated in existing models. So, accordingly a conceptual high-level model was drawn. Various modelling techniques were studies such as Abstract, Blocks, Mathematical, Circular, Spiral and Hexa models. Going ahead, we will build model progressing using one or combinations of the techniques. A dedicated brainstorming was conducted to decide, apart from the modelling techniques, what other elements should be part of the model. We concluded that, the model should also include the Deliverables, Activities, Roles of team members, Ceremonies, Techniques and Flow.

5.2 Model Building

The model was build using progressive model building technique as shown in figure 5.2. In step 1, it starts with overall flow of building a product. Here the process start with exploratory work of problem followed by defining what problem to be exactly solved. We also need to make sure that we are solving the right problem and problem will be desired by the customer. Then we set-up the hypothesis, assumption and find the very initial solution to the problem at hand and test the solution quickly.

Once our initial solution is desirable, feasible and viable, we start full-fledge product development process. Finally, we explore opportunities to grow the business further. All the potential stages groups the subtasks as highlighted in different colors in figure 5.2.

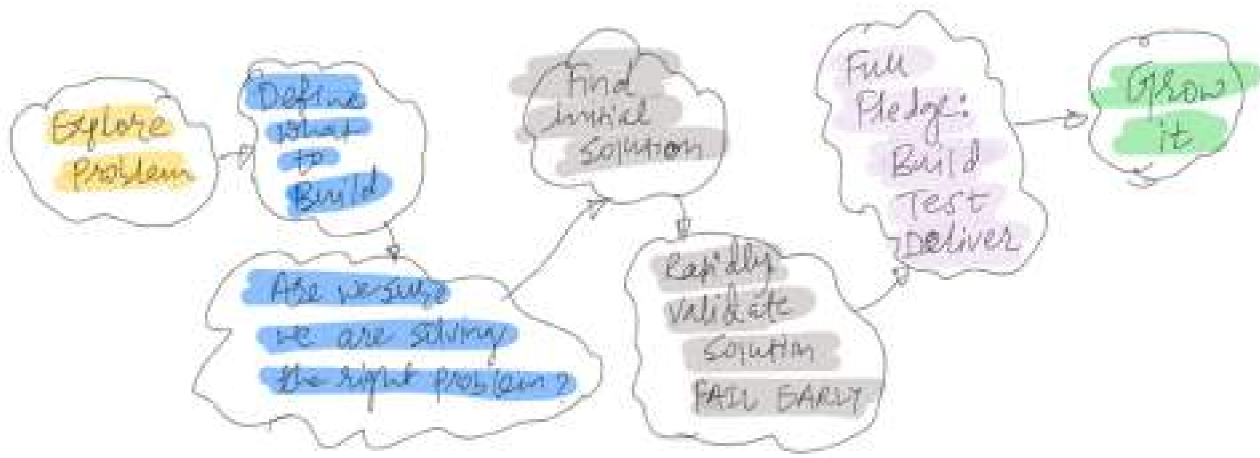


Figure 5.2 Step 1 – Overall logic flow of building solution/ product/ business

Then in step 2, we progressively added the methodologies for each of the potential stage as shown in figure 5.3. The process starts with problem exploration, in which, the focused view helps to uncover the user needs. Design Thinking (DT) technique is suitable for this purpose. For broader level view of the ecosystem, System Thinking (ST) methodology is added. Once the needs are explored, the team should focus on defining exact problem to be solved. A combination of design thinking and Lean Start-up (LS) philosophies are used in this phase. Once the problem is defined, the initial solutions, hypothesis and assumption are rapidly validated.

We also need to make sure that we are solving the right problem i.e. the solution offered should solve customers problem and it should be desired by the customer. For this, in the third phase Lean Start-up (LS) methodologies are used along with balance of other methods. The Build-Measure-Learn loop of lean Start-up helps to rapidly validate the business model, solutions. Then once the feasibility, desirability and viability of the solution is tested, a full development cycle is needed to deliver the solution product to the customer. For this Agile Methodologies (AM) are best approach to deliver efficiently, iteratively.

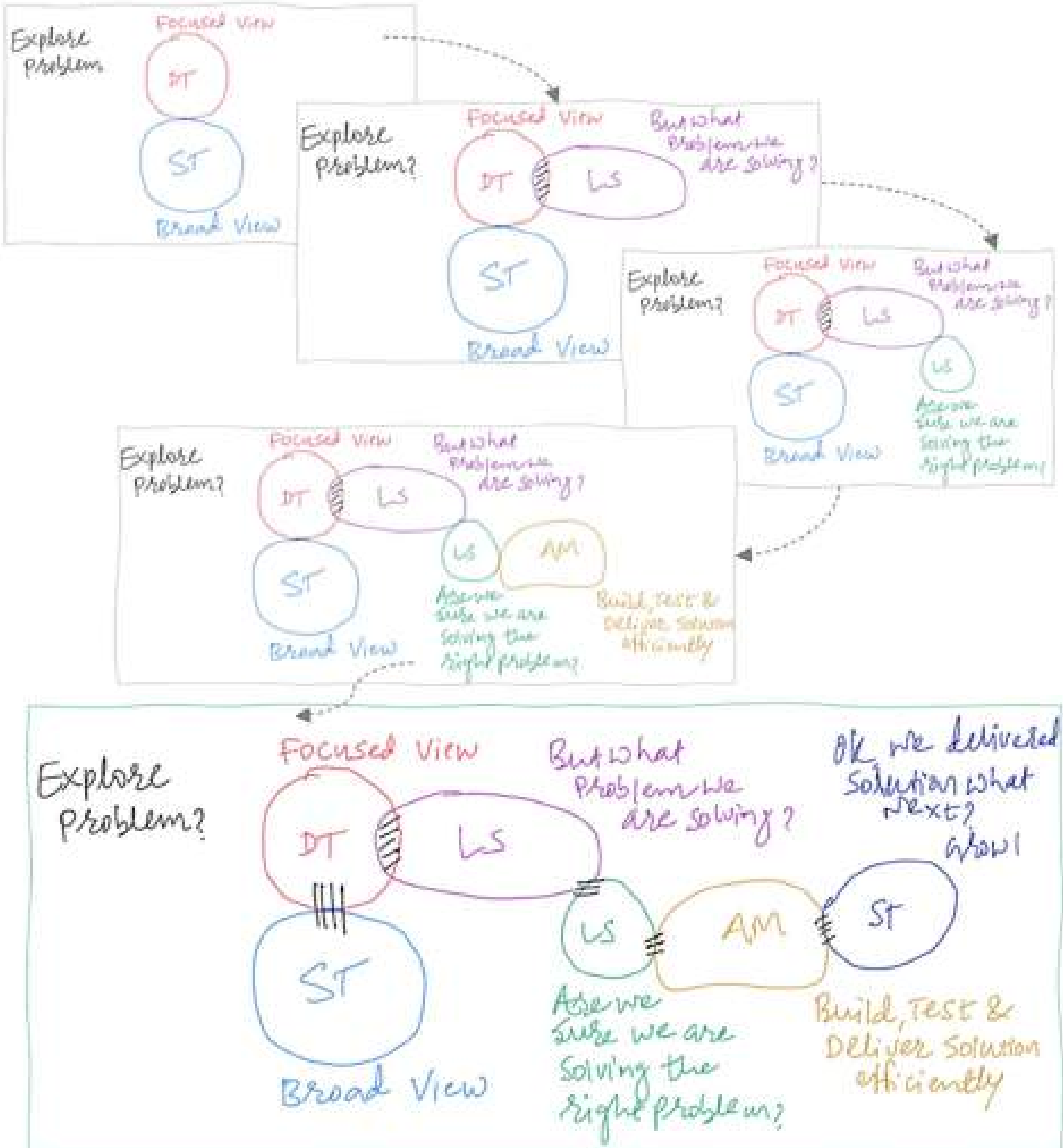


Figure 5.3 Step 2 – Progressive Modelling Work

Finally, the methodologies are logically grouped and phases are named for each group in step 3 of stage grouping, as shown in figure 5.4. The very first stage which includes Design Thinking and System Thinking is 'Discover'. The second stage for problem definition which uses the Design Thinking and Lean Start-up is named as 'Define and Validate'. The third phase is 'Incubate', which involves lean start-up with balance of other methodologies. The fourth phase is 'Iterate' which predominantly use agile methodologies to deliver the product incrementally to customer. The final stage is 'Systemize' to grow and scale-up the business further.

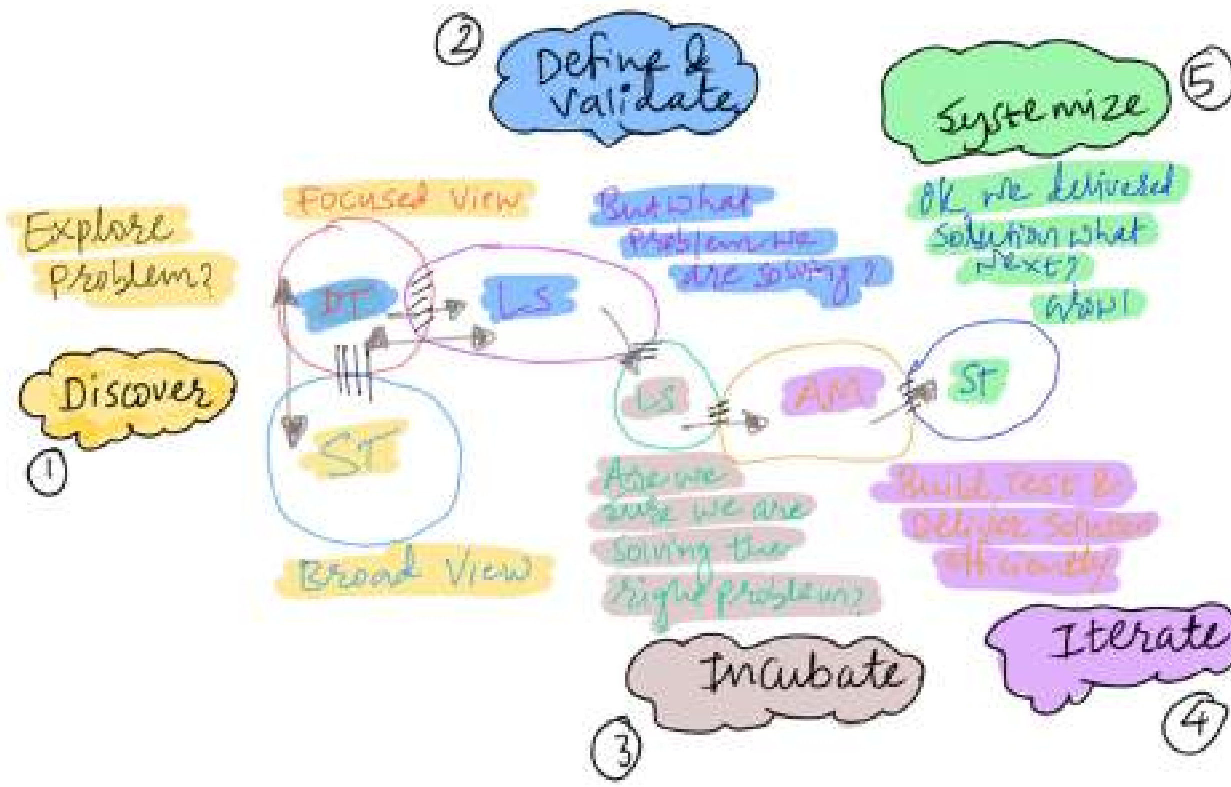


Figure 5.4 Step 3 – Stage Grouping

5.3 Lean Agile Design (LAD) Model

By using the progressive model building technique as explained in previous section, we defined the five key stages of model. These five stages when performed in logical sequence iteratively constitute the Lean Agile Design (LAD) model. The LAD model is presented at two levels – Abstract and Block. The Abstract model is shown in figure 5.5. The infinity loop encompassing the five phases indicates the agility, speed and non-waterfall nature of the model.

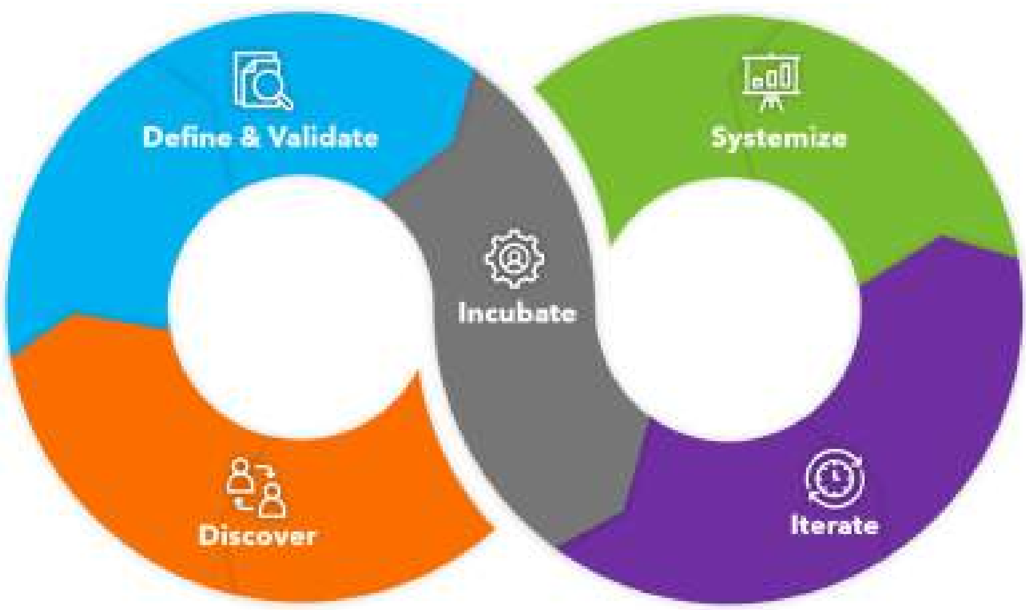


Figure 5.5 Lean Agile Design (LAD) - Abstract Model

A detailed model in form blocks and connecting flows, as shown in figure 5.6, was also developed to show the flow of the various stages of lean Agile Design model. The box model is created to describe the stages and flow in a simple manner and it is neither a waterfall model nor indicates the sequential nature of activities.

The very first 'Discover' phase is predominantly for understanding the opportunity landscape, appreciate the larger ecosystem and uncover user needs. The focus here not just on understanding the customer needs and wants, but also to have system level overview of the product offering, customer problems and market opportunity. So, Discover is the first event in creation of problem space. The methodologies used here are predominantly System Thinking (ST) and Design Thinking (DT).

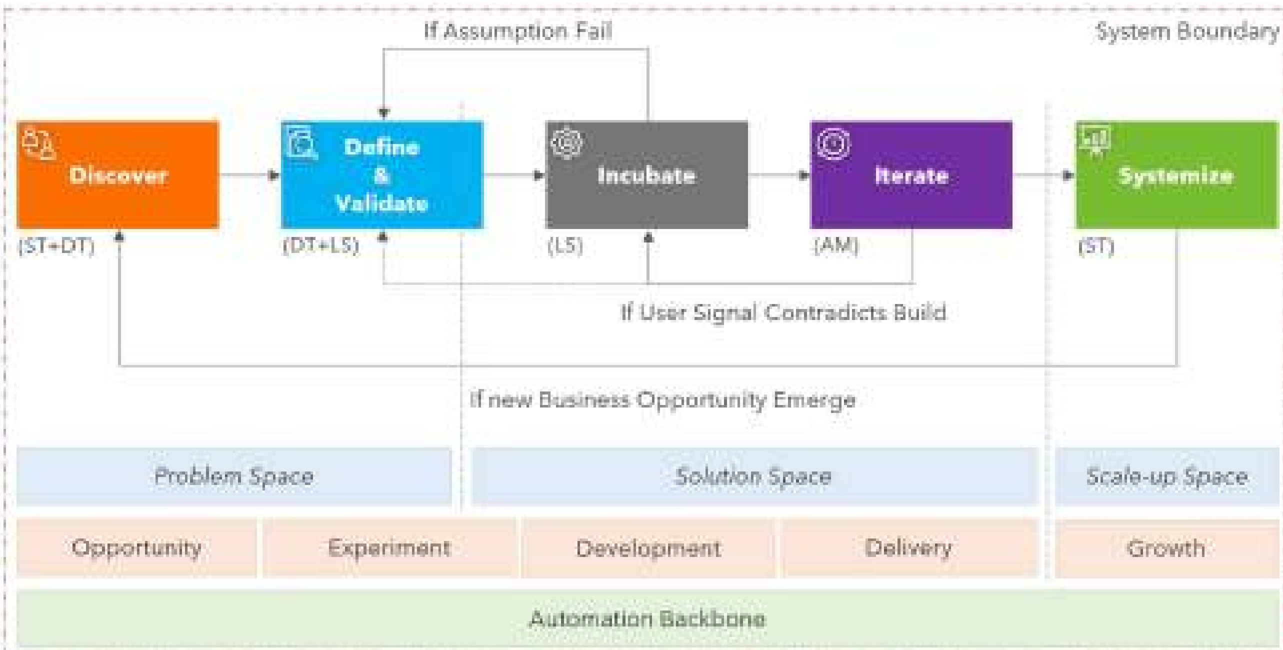


Figure 5.6 Lean Agile Design (LAD) – Block Model

The second phase 'Define and Validate' is for converging the study to define the exact problem, and test and validate the solution assumption rapidly. This is the second phase of the problem space, which focuses on rapid experimentation to mitigate early risks on potential solution to the customer problem. This phase predominantly use Design thinking (DT) Lean Start-up (LS) philosophy.

The third phase is 'Incubate', which is for creation and validation of Business Models, analyze product-market fit and refining product vision. It uses Lean Start-up (LS) philosophies such as Build-Measure-Learn (BML) loop. This is the first phase of the solution phase and focuses on initial product development work. This helps to test the hypothesis to make sure that, the product we are going build and porductionize is what needed by customer to satisfy their needs.

The fourth phase 'Iterate' starts when the product requirements are clearly defined and broken into product backlog items. The purpose of this phase is to deliver the working software pieces incrementally and agilely in form of sprint to the customer, which adds incremental value to the final product. In this phase, the full-fledge development work is conducted to develop, build, test and deploy the product. Various Agile Methodologies (AM) are used to perform the development work in this phase.

The final phase is 'Systemize', which is for reflecting on the product delivered to customer, retrospect and integrate lessons back to the programs, continually enhance the product and process performance, Scale-up and grow the business. This phase predominantly make use of System Thinking (ST) philosophy and forms the Scale-up phase. Another objective of this phase is to build resilience, and maintain & monitor product health.

Each phase has a dominance of a particular methodology, however, almost all the times a balance of all the activities is introduced in each phase. Automation is the backbone of the whole end-to-end process, which increases the speed and accuracy of the deliverables. There are also feedback loops introduced such while testing the hypothesis, if assumption fails in the Incubate phase, the team will go back to previous Define and Validate phase to redefine the problem, assumption and quickly test them. In Iterate phase, while building and testing the product, if user signal (feedback in form of data, behavior etc.) indicates mismatch with the built product, the team will go back to the Incubate phase to rerun the BML loop. The team may also go back, if there is a significant contradiction between user signal and build, to the Define and Validate phase to redefine the problem and assumptions. While scaling-up, if a new opportunity emerges, the team will go back to very first Discover phase and start a fresh program to seek the business opportunity.

5.3.1 Discover

The Discover phase of LAD model focuses on understanding opportunity landscape, finding ana validating real problems worth solving. Team moves beyond the assumption to frame the problem systematically. The focus of this phase is to find value in opportunities and for that purpose, team generate hypotheses and initiate activities which creates value in futures such as Value Proposition Canvas, Revenue model, Cost saving opportunities, initial risk reduction plan.

All opportunities are added to the opportunity backlog and priorities based on the criteria shown in figure 5.7. The desirability deals with what customer need, generating and quickly testing the idea and hypotheses. The Feasibility is analysis of opportunities for technical feasibility point of view. The Viability is about checking whether the opportunity is financially viable. And finally, the Conductability is initial check on whether the idea is scalable, sustainable, able to grow in future and ethical. At the end, the team creates a minimal, validated sets of artifacts. The success is a list hypotheses backed with evidence and clear next steps in the project.

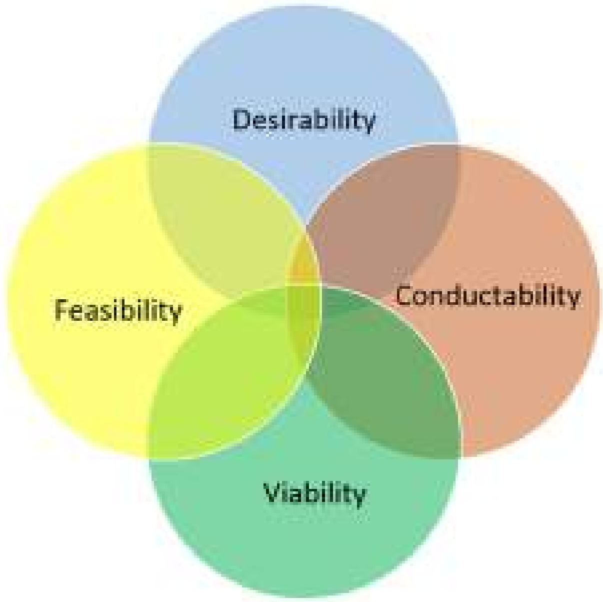


Figure 5.7 Opportunity Backlog Prioritization Venn

Deliverables:

We have categories the deliverables of discover phase into two types: Evidence-based and Decision-based.

The various evidence-based deliverables from discover phase are stakeholder map, ecosystem map, user personas & primary user models, customer journey maps and experience maps, competitive analysis & market analysis summary document, initial technical concepts (system diagrams, APIs, data availability, current architecture constraints etc., ethics & regulatory scans.

On the other hand, the decision-based deliverables of discover phase are problem statement, opportunity statements, Lean Canvas for the top opportunities, prioritized hypothesis backlog, initial definition of success metrics & acceptance criteria, low-fidelity prototype(s), initial, risk register & assumptions list, and finally, the Go/No-Go recommendation and roadmap of next-steps.

Tools & Techniques:

Various techniques used in discover phase are contextual inquiry, user interviews, initial usability testing, affinity diagrams, root-cause analysis, causal loop diagrams, systems mapping, Lean Canvas, Value Proposition Canvas (VPC), Impact vs Effort matrix, exploratory data analysis, cohort analysis, funnel/conversion analysis, segmentation, rapid prototyping, privacy impact assessment, ethical checklist and stakeholder sign-off document.

Roles:

Project sponsor provides strategic direction, funding, key decision, removes roadblocks, and align the project with organization goal tree. Product manager owns the opportunity backlog, decision-based deliverables, prioritize hypotheses and manage scope. UX Researcher conduct research on customer's usage, quantify insights, and create customer journey map and personas. UX Designer creates wireframe, rapid prototypes, interaction flow and run the design workshops. Data Analyst & Data Scientist conduct analytics on data & share actionable insights, perform cohort and funnel analysis, conduct Design of Experiments (DOE) and statistical analysis. Engineering Leads (Engineers and Architect) conducts quick product and technical feasibility check, identify any technical constraints, and estimate investment needs. DevOps Engineers and Site Reliability Engineer (SRE) calculates infrastructure requitements. Scrum Master facilitates discovery workshop, drive decision and collaborate with cross functional teams.

Ceremonies:

The Discover phase is run as a time-boxed sprint, but with flexibility of research project considering its exploratory nature. Various ceremonies and cadence include

Customer meetings, Kick-off meeting with cross functional team. The daily stand-ups are for daily progress review, removing roadblocks, and collaboration across team. Research sprint planning is for planning the two week's activities such as interviews, prototyping activities, and data analysis work. Empathy workshops are conducted after user interviews for collating the customer voice, and creating customer personas and journey maps. Insights and Synthesis review is conducted somewhere mid of the sprint in which, data is analyzed and user patterns and product themes are reviewed. Then Ideation workshops are carried to generate solution ideas for the defined high-level problem. Decision gate and steering committee review is conducted for presenting the decision deck to leadership team and get decision on project continuity. At the end, a retrospective session is conducted to understand what went right and what went wrong, collect lessons learnt, and integrating lessons into the plan for next steps.

5.3.2 Define and Validate

The Define and Validate phase transforms the prioritized hypotheses from Discover phase into clearly defined testable potential solutions. The clarity of definition helps to increase certainty across desirability, feasibility, viability and conductability of product for customer. The focus in this phase is to create a Minimum Viable Solution (MVS) which will further developed into Minimum Viable Product (MVP) in next phase. The define success metric against which experiments are also designed to measure impact of solution on metrics. This helps in conducting technical feasibility of the solution, needs for integration to mainstream application, cost considerations, security, compliance and any regulatory requirements early. A dedicated session is also run on analyzing and mitigation planning on potential risks. And finally, a dedicated exit review to make a decision on go or kill the project is conducted with steering committee members.

Deliverables:

The team creates a Solution Definition Document (SDD) to depicts the concise specification of the potential solution (MVS), which includes, technical scope, workflows, User Experience (UX)artifacts, acceptance criteria, and it may also include results of very initial Low Fidelity (Lo-Fi) mockups. The team plans experiments such as initial Verification and Validation (V&V) test plans, hypothesis testing plan, detailed A/B test plans etc. Depending on the complexity of the product solution, the team also choose to build Lo-Fi prototypes such as clickable prototypes, interactive mockups, or a small Proof of Concept (POC). The team also initiates to plan and define the instrumentation and test data schemas, logging & monitoring requirements, dashboard mockup plans etc.

Based on feasibility assessment, the estimation work is performed for engineering resources, infrastructure capacity need, test facility requirements and prototype sample requirements. A business case is initiated, which includes the inputs costs – working capital and Capital Expenditure (CAPEX), sales projects, Return on Investment (ROI), Internal Rate of Return (IRR), and Payback period (PB) to asses the financial feasibility of the project. The Value Package Canvas (VPC) is further matured in this phase as the team has better clarity on product solution. Finally, a backlog is created for user stories, Non-Functional Requirements (NFRs) and acceptance test for upcoming MVP.

Tools & Techniques:

The core technique used in Design and prototyping are design sprints, design studio work, paper prototyping, Figma, InVision, Wizard of Oz (WOz) etc. technical validation tools such as integration tests, load tests, Proof-of-concept code spikes, API mocking (such as WireMock) tools. On Experiments and analysis, for conducting A/B testing, tools such Optimizely can be used, for analytics work tool such as Mixpanel and Amplitude can be used for deep analytical work, Cohort analysis and G*Power tool for sample size calculation & statistical analysis can be used.

For qualitative testing, the team can perform moderated usability testing, Card Sorting technique, cognitive walkthroughs, think-aloud protocols and follow-up interviews with users. The team also initiate the risk and compliance work using techniques such as Privacy Impact Assessment (PIA), Sustainability Impact Assessment (SIA) for assessment of environmental, social and economic impact of project, Threat modeling technique for cybersecurity and formal legal compliance checklist.

Roles:

Product Manager leads the solution definition initiative by prioritizing validation experiments and creates a decision pack and roadmap. Design Lead creates Hi-Fi prototypes and mockups, defines acceptance criteria for UX, and provide inputs for designing usability tests. UX Researcher design and perform user tests, collate the test feedback and perform experiment moderation. After that, the Data Analyst analyze the tests results, design experiments, present insights from data in form of dashboards.

DevOps Engineers and Site Reliability Engineer (SRE) ensure infrastructure readiness and monitoring systems. Legal, Ethics and Compliance lead fills checklist and sign-off the risk mitigation plan. The Project sponsor ensure strategic alignment and gets the decision on gate. Scrum Master coordinate various ceremonies, lead teams, removes roadblocks, track cadence and makes sure the proper documentation.

Ceremonies:

The Define and Validate phase is run as a time-boxed sprint of 1-3 weeks each which helps in moving from hypothesis design to rapid protos, experiment and define project roadmap. It starts with a phase kick-off workshop where scope is clearly discussed with team and their questions are answered. The scope of target MVS is discussed and team is aligned with success metric, assumptions, constraint and risks. Design sprints are performed for rapid ideation, quick design and selection of prototypes. Daily Stand-ups (DSUs) are 15 minutes agile team meetings for discussion on brief updates and plan for roadblock removal. Mid-Experiment Touchbase is lightweight ceremony for review of telemetry and qualitative data. Decision gates are for review, stakeholder alignment and decision to go ahead in project – Pivot or Preserve. Retrospective meeting is for collecting lessons learnt from the phase. And finally, Handover planning meeting is for backlog refinement and handover the prioritized backlog, solution definition document, and experimentation outputs to development team.

5.3.3 Incubate

The Incubate phase acts as a bridge, but with an important role, between validated solution definition (from Define and Validate phase) and full-scale development and delivery (Iterate phase). In incubate, the Minimum Viable Solutions (MVS) are turned into Minimum Viable Products (MVP) so as to reduce the project's operational and integration risks. The MVP includes the validates and tested UI concepts and key valuable features. In this, the learnings from design experiments are converted into robust capabilities by proving scalability and readiness for real-world usage with an option to pivot or preserve.

The primary objective of Incubate is to generate validate concepts into a strong pilot implementation which is scalable and can be taken ahead for final product development cycle. The created MVPs also include the Non-Functional Requirements (NFRs) with optimum degree. The assumptions from previous phases are validated under realistic future load. The product vision is refined and Business Model Canvas (BMC) is created and validated. A full-scale telemetry for data collection, monitoring and dashboarding is created and implemented.

Deliverables:

The technical deliverables include the production-grade MVP codebase in source, full implementation of Continuous Integration and Continuous Delivery/ Deployment (CI/ CD) pipeline, and Code review mechanism established. Various deployment artifacts such as Docker images, Helm Charts, Infrastructure as a Code (IaC) (Terraform) templates are created. Test Automation suits are established for unit, integration, smoke and integration tests. The various operational and observability deliverables are instrumentations to collect data, dashboards, alerting rules, escalation procedure, definitions of Service Level Agreement (SLA), Service Level Objectives (SLO), Canary Release and Circuit breaker Strategy for feature releases.

For security, the PIA is updates, compliance checklist is updated and security scans are established. Business Model Canvas (BMC) is created and validated with cross functional team (CFT). The product backlog and acceptance criteria are refined.

Tools & Techniques:

The various engineering tools used for building CI/CD pipeline are Git workflows, GitHub Actions, Jenkins, Containerization & orchestration (Docker, Kubernetes, Helm), IaC (Terraform, CloudFormation). For observability, various tools are used such as Prometheus, Grafana, Datadog for metrics and monitoring; OpenTelemetry, Jaeger, Zipkin for Tracing; CloudWatch, Stackdriver for data logging; and APM and RUM for end-user performance. For quality testing Pytest, JUnit for unit & integration frameworks test; Pact for contract testing (Pact); Cypress, Playwright for end-to-end testing; JMeter and K6 for load testing. SonarQube, Snyk, OWASP ZAP are used for security and compliance testing. For feature release and control, the LaunchDarkly and Unleash are used for feature flag systems; Flagger is used for Canary deployment tooling. Various tools such JIRA, Azure DevOps, AWS are used as agile management tool. And finally for software project documentation, Confluence and notion are used.

Roles:

Product owners ensure that the pilot scope is aligned to the product goal. Engineering lead performs technical product hardening, makes product architecture decisions and refine engineering capacity. DevOps Engineers/ Platform engineers/ SRE builds CI/CD pipeline and automate infrastructure and monitoring system. Software Quality Assurance (SQA) engineers design test automation, plan tests specification based on Test Driven Development (TDD) strategy, load tests and performance test suits. Security engineer performs security scans, compliance checks and privacy assessment. Release engineer manage deployments and version control.

UX Product engineers make sure that the MVP meets design & usability specifications, collects and act on pilot feedback. Data Engineer performs full scale implementation of data telemetry, validate data schemas and analyze data with respect to pilot metrics. Business, sales and Ops SMEs validate the business case and builds the BMC. Project sponsor get the approve on pilot targets and continuity decisions.

Ceremonies:

Incubate is performed in form on few sets of sprints of 1 to 3 weeks each. The very first ceremony is Incubate kick-off, in which the pilot scope, KPIs, team structure, roles and time plan are defined. Daily Stand-up are for progress review and roadblock removal.

SRE Touchbase is for reviewing and finalizing the infra, telemetry and operational readiness. Security Compliance Synch is a meeting for reviewing and taking action on outstanding points related to security. Pilot War Room meetings is for resolving initial inflow of issues post pilot launch using Gemba. And finally, the Retrospective session which support the lean philosophies' BML cycle includes the brainstorming session n lessons from the phase and creating plan to avoid repetition of issues.

5.3.4 Iterate

The Iterate phases involves launching a full-scale agile product development work to deliver the values added product iteratively to customer. The agile team works on building product by prioritizing user stories, test, deploy and productionize the product. The software product is not delivered all at once, but delivered to customer in form of pieces which includes minimum working features which adds incremental value to final application. This helps in getting early feedback on the product performance thereby supporting on time launch and minimizing the rework.

Deliverables:

The main deliverable of this phase is working software delivered to customer. The Epics formation, user stories definition, story size estimation and defining non-functional requirements are few of the initial deliverables of Iterate. The product backlog is prioritized and refined by the team. The Burn-down chart and velocity tracking are done for measuring the performance of the sprint and product development process. And finally, the technical documentation is also an important deliverable.

Tools & Techniques:

The Program Board and Kanban boards (JIRA, Trello, Azure DevOps) are used for managing the user stories, workflows, and work management during product development. The documents related user stories acceptance criteria are defined in advance. Test Automation tools such as Selenium, JUnit, Cypress are used for performing tests automatically on the software code. The DevOps tools such as Docker, Jenkins, GitLab CI etc. are used in this phase as a part of CI/CD pipeline. Various agile methodologies such as pair Programming, TDD (Test-Driven Development) and BDD (Behavior-Driven Development) are also used during agile execution.

Roles:

The Product Owner (PO) is the owner of product backlog, who is responsible for maximizing the value of the product in each iteration. The PO defines vision, prioritize the Product Backlog Items (PBIs), prioritize feature delivery, and aligns the product with business goals. The Scrum Master (SM) is a leadership role who coach agile team, promote and ensure Scrum or Scaled Agile framework (SAFe) in project team, remove impediments, facilitate scrum ceremonies, and protect project team from external distractions.

The Development Team consists of developers for frontend and backend work, who design, code, maintain the software application, collaborate with other teams and debug codes. The Software Quality Assurance (SQA) engineers are responsible for ensuring the software product meets pre-defined quality standard and user needs. The SQA engineer works closely with developers to create a test plan, documents defects, and clearly provides feedback to developer. The UI/ UX engineers while working on design research and testing usability, make sure that a usable and valuable product is delivered to the customer. The

DevOps engineer connects the development and operations team to speed up the process and enhance efficiently of deployment and production.

Ceremonies:

The PI (Program Increment) Planning is the main event at the start, in which all teams from Agile Release Train (ART) come together to align in objective, identify dependencies and create a plan with common understanding. The Iteration planning is a team level event to define stories, tasks, create and prioritize stories for immediate sprint. The Daily Stand-up meeting is a short everyday meeting to discuss on yesterday's progress, remove roadblocks and impediments, and plan for upcoming days. Iteration review is the review of the work done in the iteration and receive feedback on product demo from the relevant team member. The Iteration retrospective is the review and analyze the lessons learnt from past iteration and create improvement plan on those.

The System Demo is the event in which the team review the integrated product from various teams. The Scrum of Scrums (SoS) is the meeting in which the key representative such as Product Owner, Scrum Masters, key developers of various agile teams meet and discuss about dependencies between teams and identify roadblocks. The PO Sync is the meeting in which the POs of various teams meet and discuss about the backlog priorities based on program scenario. The Inspect & Adapt (I&A) is a major event at the end of Program Increment (PI), in which the whole ART teams demonstrate the complete product. The Portfolio Sync is the event in which the leadership team comes together to review the progress of program portfolio with respect to business objective, align on strategic initiatives and assign resources and budget.

5.3.5 Systemize

The key purpose of the Systemize phase is to scale and grow operationally and integrate the feedback from ecosystem. In this phase, the team build the resilience, scale-up, grow, maintain and improve product health. Once the ideas are discovered, defined, validated, incubated, iterated and delivered, the Systemize phase once success into a long-term system by implementing Standard Operating Procedure (SOP), Toolchain automation, product platform scalability, strong governance, continuous improvement loops, organizational skill development, cost improvement, improving compliance and sustainability. The growth aspect is in terms of increasing user volumes, team adoption, technical scale-up. The sustainability aspect is on making sure the long term ethical, environmental, operational, financial and regulatory sustainable business. This help to make the one-time success from iterate phase more repeatable, scalable and predictable in future.

Deliverables:

The production grade, versioned, automated platform with modules such as CI/ CD, IaC, curated containerization etc. to build high concurrency on product usage. Building autoscaling templates, multi-region deployment of product for seeking users from new regions, and Disaster Recovery (DR) Rulebook for recovering system after any outage or disaster during scale-up.

With user base growth, the operational and observability stacks also need scaling up. The monitoring stacks such as metric, log pipes scaled up with roll-up metrics and enhanced retention policies. Capacity plans and Service level Objectives (SLO) as per sales growth forecast. The enhanced monitoring dashboards such more multi-level, team-level, region-level details with more slice and dice features.

With increased user base, on sustainability and compliance, the artifacts are delivered such as Data lifecycle policies, Energy or CO_2 reporting for increased cloud usage, achieving Carbon Credits, and templates submission for regulatory compliance. Growth governance deck is released for approval on increased flow, capacity enhancement, cross-regional roll-outs, and feature which impacts revenue growth.

Business case update with unit economic model aspect such Customer Acquisition Cost (CAC), Customer Lifetime Value (LTV), cost margin per customer etc. Cost optimization playbook for avenues to reduce cost such as cloud usage, Caching, Tiered storages, Data compression etc. People onboarding plans, training plans for increased workforce, community establishment for focus on continuous learning, scale-up, Frugal engineering and Green engineering.

Tools & Techniques:

For enhanced capacity and cost modelling various tools are used such as time series forecasting and scenarios simulation/ Monte Carlo simulation are used. For cloud cost calculation, forecasting and management, tools such as Cloud cost APIs and FinOps tooling are used. For observability, various tools such as OpenTelemetry, metric roll-ups, Low cardinality schemas are used.

For requirement of enhanced governance and policy, engine such as OPA (Open Policy Agent) with policy-as-code are deployed, and checks are introduced in CI/ CD pipeline to meet enhanced capacity and budgets. For sustainable engineering practices, various techniques such as right region selection, workload balancing and scheduling, and Recommender APIs for rightsizing recommendation on tech resource management. On FinOps (Financial Operations), various techniques are used such as cost allocation, budget alert workflow, spend anomaly detection, cost optimization to optimize cloud & resources cost and maximize business value.

Roles:

The Platform Product Manager acts as a sustainability and scalability lead in this phase, who owns the platform roadmap for targets and track with respect to the forecasted capacity and cost. Siter Reliability Engineer (SRE) and Platform Engineer builds auto-scalable infrastructure and own the infra capacity planning. FinOps Specialist or Cloud Economist owns the spends, chargebacks, cost optimization, and tradeoffs related to cloud operations. Data Engineer implement the scalable data telemetry pipeline. Security and compliance lead take the privacy and regulatory constraints across new regions. The Product Manager sets the vision, growth targets, KPIs and optimize the capacity and budget tradeoff. Sustainability officer analyze and recommend the low carbon technical choices, track the sustainability metrics, and makes sure that the sustainability goals are achieved. The Developer Experience (DevEx) Team helps the development team to adapt to the grown-up platform and processes.

Ceremonies:

Platform Stand-up (PSU) Meeting every week is organized for reviewing cost vs capacity snapshot and identify spend anomalies. Growth Planning Sync-up meeting is done fortnightly to review the product roadmap and its alignment with capacity plan, and how the company is adopting to the product market growth.

Monthly meeting on Cost and Capacity review for analyzing trends, forecasts and identifying optimization opportunities. There are monthly discussion on incident and SLO reviews. Quarterly reviews with leadership team for readiness of company for scaling, approving expansion plans, features and platforms. Sustainability review is done quarterly to review the carbon credits, CO_2 emission status, energy efficiency and organizational compliance status.

5.4 Conclusion

In this chapter, we analyzed and concluded that, there are top, most influential seven independent variables which governs the project success. We conducted a deep-thinking session on whiteboard and defined a model building strategy, technique and all related aspect. Then by using a progressive mode building technique we built the lean Agile Design (LAD) model. There are five phases in LAD model which are not in sequence but they are performed more iteratively and incrementally. We then described these five phases with help of overall purpose, key deliverables, techniques used, various roles involved and ceremonies conducted. Overall, the model is well defined and very useful for developing and delivering a software product very effectively and efficiently.

CHAPTER VI

SUMMARY, IMPLICATIONS, AND RECOMMENDATIONS

6.1 Summary

The research started with questions such as what are governing variables and their relationship with software development project success. We conducted a detailed literature review using Systematic Integrated Mapping Method (SIMM) methodology and found a list of pitfalls and research gap in current process models. Then we collected and analyzed the industry real project data on functional effort variance and schedule variance. We also conducted an in-depth study of various product development methods in industry. From the insights from these studies, we created a list of sub-dependent variable (yi) and independent variable (xi). For all sub-dependent variables, we defined a main dependent variable – project success (Y).

Then we formed a questionnaire for each of the independent and sb-dependent variables and conducted a survey for required sample size. Then we performed cleaning, coding, rationality analysis on the data. We conducted a Principal Component Analysis (PCA) on the data and found the most significant component, which we called as project success (Y). The dataset with the principal component was used for performing the Exploratory Data Analysis (EDA) and initial insights formation. Then we built 15 models with variety from statistical models, ANOVA, Regression models to advance Machine Learning models. The models helped to meet objectives such predicting the performance of the project for given input variables and understanding the most significant independent variables which governs the project success (Y). We checked the frequency of occurrence of the variables across all models and measured the performance of the variables. Based on the detailed study, it was observed that, there are seven variables which governs project success as x_1_Empathy, x_2_Automation, x_4_SustEthics, x_6_Growth, x_7_Human,

x_9_Governance and x_{12}_Retrospect. The, we have also built a mathematical equation to consider effect of non-linearity, variable interaction effect and saturation.

Further using Factors analysis, Deep-thinking exercise and Progressive model building methods, we built a full-scale end-to-end software product development model – Lean Agile Design (LAD). The model consisted of five phases such as Discover, Define & Validate, Incubate, Iterate and Systemize. These phases are not in waterfall sequence, but they are agile, iterative and overlapping in nature like an infinity loop. For simplicity of the explanation, we represented the model using blocks. This model uses the combination of techniques such as design thinking, system thinking, lean start-up, agile methodologies, and automation.

6.2 Implications

The Lean Agile Design (LAD) is a cohesive model for software development projects which overcomes various challenges in current methodologies such as fragmentation, handling complexity, predictability, automation etc. From detailed systematic literature review and research, it was observed that, there many pitfalls in current models and there has been insufficient work done to build an integrated model which can arrest the observed pitfalls. Our research integrates all the critical learning from the current research, industry data analysis, and voices from industry practitioners.

Our model is built on fragmented strengths of various philosophies such as design thinking, system thinking, lean start-up, agile methodologies, and automation into one integrated end-to-end flow. This model emphasizes that, the software product development engineering is not only operated by speed and accuracy, but also by flexibility, sustainability and systemic balance. The finding from the statistical and machine Learning models such clarity of vision, empathy with customer, automation, learning loops, Human-

Centered Design, governance, scalability and growth, are aligned with the model flow. This results into data driven decision making, frequent & continuous learning and feedback loops, predictable project success, alignment of people, process and systems, and business value.

Apart from the governing factors and LAD model, we also built the machine learning models which have a good performance and accuracy on predicting the software development project success for given project parameters. From academic point of view, the research forms a theory of synthesizing and integrating various multi-disciplinary models into one cohesive meta model. It also supports the advancement in understanding and application of advanced new generation technologies to building models in software engineering. For industry, it provides scalable and ready to use AI predictive models for predicting project success and a software development model for actually building and delivering software to customer. So, in summary, this research work creates a research-to-practice continuum, which contribute to industry practitioners and academicians with scientifically validated approach to developing ultra-complex software products.

6.3 LAD Model Implementation Strategy

The LAD model implementation as a product development methodology in an organization will start with mindset and cultural shift. The teams need to think from systemic point of view rather than only team velocity. The leaders should think of a project as a living system rather than only a dry objective driven work, which will help in inculcate the empathy, usability, Human-centered Deign, learning and governance in the project teams. The top-level management should establish an LAD Transformation Office (LTO) to which will include a dedicated cross-functional team responsible for LAD rollout, training, governance and continuous performance monitoring. At very start, the model should well embed into the corporate strategy by including metrics into executive dashboards, KPI, and OKRs, to have top-down push for implementation.

The implementation should start with a pilot program which cross-functional, highly visible and complex enough to fetch the value from LAD model. The roles should be clearly defined, the deliverables should be clearly communicated and the ceremonies should be marked on team's calendar with right cadence. While implementing governance, a care should be taken that, the model should have enough flexibility and agility to customize it as per organizational and project needs. Certain aspects of models should be institutionalized such as feedback loops, data driven decisions, empathy with customer, early testing etc. While implementing, the process automation using techniques such as DevOps should be integrated early in the project to promote zero-touch atmosphere and enhancing speed and accuracy of the deliverables. The tech-stack such as new infrastructure, cloud, networks, tools, CI/CD pipeline, telemetry, AI tools should be planned at the start of implementation. Integrate sustainability metrics, process standardization, continuous learning culture in the day-to-day work. And finally, the team should encourage cross industry collaborations and alignment with policies.

6.4 Recommendations for Future Research

In this research, we built the various models from industry data and insights from voices of industry practitioners, followed a full-scale LAD model for complex software development methodology. Going ahead, the research can be extended to actual usage and implementation of the model in software development companies. This will help to further establish the strengths of the model and integrate learnings back into the model flow. The feedback from actual usage will also definitely help to compare the model efficacy with respect to those of existing models such as Scrum, SAFe, XP, Kanban and other specific models used in organizations. The efficacy should be measured on performance on overall project success and all the sub-dependent variables.

We have developed this model considering complex software product development projects, which can be extended to similar technology development programs. However, the model can also be applied to different products and organizations to have a new view of applicability and customization requirements.

6.5 Conclusion

This research has resulted into various AI models which have capability to predict the performance of real time software development projects by providing inputs on independent variables. The performance and accuracy of these models very high and can be applied to industrial projects. The systematic literature review also resulted into deep insights on takeaways from current literature, gap and list of pitfalls in current software development processes. An advanced mathematical equation was also derived to consider effect practical aspects such as non-linearity, variable interaction effect and saturation. Another key output is Lean Agile Design (LAD) framework designed based on inputs from survey data, analysis of industry data, critical review of literature and insights from various statistical and machine learning models. Overall, the research work has a contribution to the body of knowledge of software engineering, product development and project management. The models built during the course of the research can be applies to industry and academic scenarios to enhance efficiency, predictability and on-time launch of high-quality products.

REFERENCES

[1] Adikari, S., McDonald, C. and Campbell, J., 2009. Little design up-front: a design science approach to integrating usability into agile requirements engineering. In *Human-Computer Interaction. New Trends: 13th International Conference, HCI International 2009, San Diego, CA, USA, July 19-24, 2009, Proceedings, Part I 13* (pp. 549-558). Springer Berlin Heidelberg.

[2] Adikari, S., McDonald, C. and Campbell, J., 2013. Reframed contexts: design thinking for agile user experience design. In *Design, User Experience, and Usability. Design Philosophy, Methods, and Tools: Second International Conference, DUXU 2013, Held as Part of HCI International 2013, Las Vegas, NV, USA, July 21-26, 2013, Proceedings, Part I 2* (pp. 3-12). Springer Berlin Heidelberg.

[3] Ahmed, B., Dannhauser, T. and Philip, N., 2018, September. A lean design thinking methodology (LDTM) for machine learning and modern data projects. In *2018 10th Computer Science and Electronic Engineering (CEEC)* (pp. 11-14). IEEE.

[4] Aldave, A., Vara, J.M., Granada, D. and Marcos, E., 2019. Leveraging creativity in requirements elicitation within agile software development: A systematic literature review. *Journal of Systems and Software*, *157*, p.110396.

[5] Alhazmi, A. and Huang, S., 2020, May. Integrating design thinking into scrum framework in the context of requirements engineering management. In *Proceedings of the 3rd International Conference on Computer Science and Software Engineering* (pp. 33-45).

[6] Almeyda, S., Zapata Del Río, C. and Cohn, D., 2021. Integration of user experience and agile techniques for requirements analysis: a systematic review. In *International Conference on Human-Computer Interaction* (pp. 187-203). Springer, Cham.

[7] Ardito, C., Baldassarre, M.T., Caivano, D. and Lanzilotti, R., 2017, May. Integrating a SCRUM-based process with human centred design: an experience from an action research study. In *2017 IEEE/ACM 5th International Workshop on Conducting Empirical Studies in Industry (CESI)* (pp. 2-8). IEEE.

[8] Argumanis, D., Moquillaza, A. and Paz, F., 2020. Challenges in integrating SCRUM and the user-centered design framework: a systematic review. In *Human-Computer Interaction: 6th Iberomarican Workshop, HCI-Collab 2020, Arequipa, Peru, September 16–18, 2020, Proceedings 6* (pp. 52-62). Springer International Publishing.

[9] Aulet, B.: Disciplined Entrepreneurship: 24 Steps to a Successful Startup. Wiley, New York (2013)

[10] Babbie, E.R., 2020. *The Practice of Social Research*. 15th ed. Boston: Cengage Learning.

[11] Bailey, J., Budgen, D., Turner, M., Kitchenham, B., Brereton, P. and Linkman, S., 2007, September. Evidence relating to Object-Oriented software design: A survey. In First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007) (pp. 482-484). IEEE.

[12] Beck, K., et al., 2001. *Manifesto for Agile Software Development.* [online] Available at: https://agilemanifesto.org/

[13] Beckman, S.L. and Barry, M., 2007. Innovation as a learning process: Embedding design thinking. *California management review*, *50*(1), pp.25-56.

[14] Blank, S., 2013. Why the lean start-up changes everything. *Harvard business review*, 91(5), pp.63-72.

[15] Blosch M., Brand S., Osmond N., 2019. Enterprise Architects Combine Design Thinking, Lean Startup and Agile to Drive Digital Innovation. *Gartner Research*.

[16] Blosch, M., Osmond, N. and Norton, D., 2016. Enterprise architects combine design thinking, lean startup and agile to drive digital innovation.

[17] Boehm, B. and Turner, R.N., 2003. *Balancing agility and discipline: A guide for the perplexed*. Addison-Wesley Professional.

[18] Boehm, B.W., 2011. Software engineering economics. In *Software pioneers: Contributions to software engineering* (pp. 641-686). Berlin, Heidelberg: Springer Berlin Heidelberg.

[19] Boone, H.N. and Boone, D.A., 2012. Analyzing Likert Data. *Journal of Extension*, 50(2), Article 2TOT2.

[20] Bosch, J. and Bosch-Sijtsema, P.M., 2011. Introducing agile customer-centered development in a legacy software product line. *Software: Practice and Experience*, *41*(8), pp.871-882.

[21] Bourimi, M., Barth, T., Haake, J.M., Ueberschär, B. and Kesdogan, D., 2010. AFFINE for enforcing earlier consideration of NFRs and human factors when building socio-technical systems following agile methodologies. In *Human-Centred Software Engineering: Third International Conference, HCSE 2010, Reykjavik, Iceland, October 14-15, 2010. Proceedings 3* (pp. 182-189). Springer Berlin Heidelberg.

[22] Brad, S., Brad, E. and Homorodean, D., 2019. CALDET: a TRIZ-driven integrated software development methodology. In *New Opportunities for Innovation Breakthroughs for Developing Countries and Emerging Economies:*

19th International TRIZ Future Conference, TFC 2019, Marrakesh, Morocco, October 9–11, 2019, Proceedings 19 (pp. 400-416). Springer International Publishing.

[23] Brhel, M., Meth, H., Maedche, A. and Werder, K., 2015. Exploring principles of user-centered agile software development: A literature review. Information and software technology, 61, pp.163-181.

[24] Brown, T. and Katz, B., 2011. Change by design. *Journal of product innovation management*, *28*(3), pp.381-383.

[25] Brown, T. and Wyatt, J., 2010. Design thinking for social innovation. *Stanford Social Innovation Review*, 8(1), pp.31–35.

[26] Brown, T., Change by design: How design thinking creates new alternatives for business and society, 2009.

[27] Buchan, J., Bano, M., Zowghi, D., MacDonell, S. and Shinde, A., 2017, June. Alignment of stakeholder expectations about user involvement in agile software development. In *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering* (pp. 334-343).

[28] Burdakov, A. and Ahn, M.J., 2025. Is PMBOK Guide the Right Fit for AI? Re-evaluating Project Management in the Face of Artificial Intelligence Projects. *arXiv preprint arXiv:2506.02214*.

[29] Butler, K.A., 1996. Usability engineering turns 10. *interactions*, *3*(1), pp.58-75.

[30] Cano, S.P., González, C.S., Collazos, C.A., Arteaga, J.M. and Zapata, S., 2015. Agile software development process applied to the serious games development for children from 7 to 10 years old. *International Journal of Information Technologies and Systems Approach (IJITSA)*, *8*(2), pp.64-79.

[31] Carmines, E.G. and Zeller, R.A., 1979. *Reliability and Validity Assessment*. Beverly Hills, CA: Sage Publications.

[32] Carroll, N. and Richardson, I., 2016, May. Aligning healthcare innovation and software requirements through design thinking. In *Proceedings of the international workshop on software engineering in healthcare systems* (pp. 1-7).

[33] Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C. and Wirth, R., 1999, March. The CRISP-DM user guide. In *4th CRISP-DM SIG Workshop in Brussels in March* (Vol. 1999). sn.

[34] Chourasia, S., Tyagi, A., Pandey, S.M., Walia, R.S. and Murtaza, Q., 2022. Sustainability of Industry 6.0 in global perspective: benefits and challenges. *Mapan*, *37*(2), pp.443-452.

[35] Christensen, C.M., 2015. *The innovator's dilemma: when new technologies cause great firms to fail*. Harvard Business Review Press.

[36] Christenson, D. and Walker, D.H., 2008. Using vision as a critical success element in project management. *International Journal of Managing Projects in Business*, *1*(4), pp.611-622.
[37] Cochran, W.G., 1977. *Sampling Techniques*. 3rd ed. New York: John Wiley & Sons.
[38] Cohn, M., 2004. User stories applied: For agile software development. Addison-Wesley Professional.
[39] Cooper, R.G., 2019. The drivers of success in new-product development. *Industrial marketing management*, *76*, pp.36-47.
[40] Crawford, L.H. and Bryce, P., 2003. Project monitoring and evaluation: A method for enhancing the efficiency and effectiveness of aid project implementation. *International Journal of Project Management*, 21(5), pp.363–373.
[41] Curcio, K., Santana, R., Reinehr, S. and Malucelli, A., 2019. Usability in agile software development: A tertiary study. *Computer Standards & Interfaces*, *64*, pp.61-77.
[42] Da Silva, T.S., Martin, A., Maurer, F. and Silveira, M., 2011, August. User-centered design and agile methods: a systematic review. In 2011 AGILE conference (pp. 77-86). IEEE.
[43] Darrin, M.A.G. and Devereux, W.S., 2017, April. The Agile Manifesto, design thinking and systems engineering. In *2017 Annual IEEE International Systems Conference (SysCon)* (pp. 1-5). IEEE.
[44] De Paula, D.F. and Araújo, C.C., 2016. Pet empires: combining design thinking, lean startup and agile to learn from failure and develop a successful game in an undergraduate environment. In *HCI International 2016–Posters' Extended Abstracts: 18th International Conference, HCI International 2016, Toronto, Canada, July 17-22, 2016, Proceedings, Part I 18* (pp. 30-34). Springer International Publishing.
[45] Deeb, G.: The unlucky 13 reasons startups fail (2013). http://www.forbes.com/sites/ georgedeeb/2013/09/18/the-unlucky-13-reasons-startups-fail/
[46] Dikert, K., Paasivaara, M. and Lassenius, C., 2016. Challenges and success factors for large-scale agile transformations: A systematic literature review. *Journal of Systems and Software*, *119*, pp.87-108.
[47] Dillman, D.A., Smyth, J.D. and Christian, L.M., 2014. *Internet, Phone, Mail, and Mixed-Mode Surveys: The Tailored Design Method*. 4th ed. Hoboken: Wiley.

[48] Dobrigkeit, F. and De Paula, D., 2017. The best of three worlds-the creation of innodev a software development approach that integrates design thinking, scrum and lean startup. In *DS 87-8 Proceedings of the 21st International Conference on Engineering Design (ICED 17) Vol 8: Human Behaviour in Design, Vancouver, Canada, 21-25.08. 2017* (pp. 319-328).
[49] Dobrigkeit, F. and De Paula, D., 2019b, August. Design thinking in practice: understanding manifestations of design thinking in software engineering. In *Proceedings of the 2019 27th ACM joint meeting on European software engineering conference and symposium on the foundations of software engineering* (pp. 1059-1069).
[50] Dobrigkeit, F., De Paula, D. and Carroll, N., 2020a, November. InnoDev workshop: a one-day introduction to combining design thinking, lean startup and agile software development. In *2020 IEEE 32nd Conference on Software Engineering Education and Training (CSEE&T)* (pp. 1-10). IEEE.
[51] Dobrigkeit, F., De Paula, D. and Uflacker, M., 2019a. InnoDev: a software development methodology integrating design thinking, scrum and lean startup. *Design Thinking Research: Looking Further: Design Thinking Beyond Solution-Fixation*, pp.199-227.
[52] Dobrigkeit, F., Pajak, P., de Paula, D. and Uflacker, M., 2020b. DT@ IT toolbox: design thinking tools to support everyday software development. *Design Thinking Research: Investigating Design Team Performance*, pp.201-227.
[53] Dybå, T., Dingsøyr, T.: Empirical studies of agile software development: A systematic review. Inf. Softw. Technol. 50, 9–10, 833–859 (2008).
[54] Efeoglu, A., Møller, C., Sérié, M. and Boer, H., 2013. Design thinking: characteristics and promises. In *Proceedings 14th International CINet Conference on Business Development and Co-creation* (pp. 241-256). Continuous Innovation Network.
[55] Eris, O., 2007. Insisting on truth at the expense of conceptualization: can engineering portfolios help?. *International Journal of Engineering Education*, *22*(3), p.551.
[56] Etikan, I. et al., 2016. Comparison of convenience sampling and purposive sampling. *AJTAS*, 5(1), pp.1–4.
[57] Evans, J.R. and Mathur, A., 2005. The value of online surveys. *Internet Research*, 15(2), pp.195–219.
[58] Ferreira Martins, H., Carvalho de Oliveira Junior, A., Dias Canedo, E., Dias Kosloski, R.A., Ávila Paldês, R. and Costa Oliveira, E., 2019. Design thinking: Challenges for software requirements elicitation. *Information*, *10*(12), p.371.

[59] Ferreira, J., Sharp, H. and Robinson, H., 2011. User experience design and agile development: managing cooperation through articulation work. *Software: Practice and Experience*, *41*(9), pp.963-974.
[60] Filippov, S., Mooi, H.G., van der Weg, R. and van der Westen, L.J., 2012. Strategic alignment of the project portfolio: an empirical investigation. In *PMI Research and Education Conference 2012* (pp. 1-36). Project Management Institute.
[61] Fischer, H. and Senft, B., 2016. Human-Centered Software Engineering as a Chance to Ensure Software Quality Within the Digitization of Human Workflows. In *Human-Centered and Error-Resilient Systems Development: IFIP WG 13.2/13.5 Joint Working Conference, 6th International Conference on Human-Centered Software Engineering, HCSE 2016, and 8th International Conference on Human Error, Safety, and System Development, HESSD 2016, Stockholm, Sweden, August 29-31, 2016, Proceedings 8* (pp. 30-41). Springer International Publishing.
[62] Fitriani, W.R., Rahayu, P. and Sensuse, D.I., 2016, October. Challenges in agile software development: A systematic literature review. In *2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS)* (pp. 155-164). IEEE.
[63] Forbrig, P. and Saurin, M., 2016. Supporting the HCI aspect of agile software development by tool support for UI-pattern transformations. In *Human-Centered and Error-Resilient Systems Development: IFIP WG 13.2/13.5 Joint Working Conference, 6th International Conference on Human-Centered Software Engineering, HCSE 2016, and 8th International Conference on Human Error, Safety, and System Development, HESSD 2016, Stockholm, Sweden, August 29-31, 2016, Proceedings 8* (pp. 17-29). Springer International Publishing.
[64] Forbrig, P., 2016, April. Continuous software engineering with special emphasis on continuous business-process modeling and human-centered design. In *Proceedings of the 8th International Conference on Subject-oriented Business Process Management* (pp. 1-4).
[65] Forbrig, P., 2016. When do projects end?–the role of continuous software engineering. In *Perspectives in Business Informatics Research: 15th International Conference, BIR 2016, Prague, Czech Republic, September 15–16, 2016, Proceedings 15* (pp. 107-121). Springer International Publishing.
[66] Frauenberger, C., Rauhala, M. and Fitzpatrick, G., 2017. In-action ethics. *Interacting with computers*, *29*(2), pp.220-236.

[67] Gabriel, S., Niewoehner, N., Asmar, L., Kühn, A. and Dumitrescu, R., 2021. Integration of agile practices in the product development process of intelligent technical systems. *Procedia CIRP*, *100*, pp.427-432.
[68] Gamble, M.T., 2016, May. Can metamodels link development to design intent?. In *Proceedings of the 1st International Workshop on Bringing Architectural Design Thinking into Developers' Daily Activities* (pp. 14-17).
[69] Glomann, L., 2018. Introducing 'human-centered agile workflow'(hcaw)-an agile conception and development process model. In (Vol. 607, p. 647-655). *Springer Verlag. doi*, *10*, pp.978-3.
[70] González-González, C.S., Toledo-Delgado, P. and Muñoz-Cruz, V., 2015. Agile human centered methodologies to develop educational software. *Dyna*, *82*(193), pp.187-194.
[71] Goodman, E. and Kuniavsky, M., 2012. *Observing the user experience: A practitioner's guide to user research*. Elsevier.
[72] Goulstone, P., Aslan, K., Smith, N., Kronborg E. N.D. How fragile is your Agile? Six common pitfalls facing Agile project teams. Deloitte.
[73] Green, S.B., 2015. *The number of subjects per variable required in linear regression: A review*. International Journal of Social Research Methodology, 18(6), pp.559–568. https://doi.org/10.1080/13645579.2014.953971
[74] Griffith, E., 2014. Why startups fail, according to their founders. *Fortune Magazine, September*, *25*.
[75] Grossman-Kahn, B. and Rosensweig, R., 2012. Skip the silver bullet: driving innovation through small bets and diverse practices. *Leading Through Design*, *18*, p.815.
[76] Gurusamy, K., Srinivasaraghavan, N. and Adikari, S., 2016. An integrated framework for design thinking and agile methods for digital transformation. In *Design, User Experience, and Usability: Design Thinking and Methods: 5th International Conference, DUXU 2016, Held as Part of HCI International 2016, Toronto, Canada, July 17–22, 2016, Proceedings, Part I 5* (pp. 34-42). Springer International Publishing.
[77] Häger, F., Kowark, T., Krüger, J., Vetterli, C., Übernickel, F. and Uflacker, M., 2015. DT@ Scrum: integrating design thinking with software development processes. Design thinking research: building innovators, pp.263-289.
[78] Hanington, B. and Martin, B., 2019. *Universal methods of design expanded and revised: 125 Ways to research complex problems, develop innovative ideas, and design effective solutions*. Rockport publishers.
[79] Hehn, J. and Uebernickel, F., 2018, August. The use of design thinking for requirements engineering: an ongoing case study in the field of innovative

software-intensive systems. In *2018 IEEE 26th international requirements engineering conference (RE)* (pp. 400-405). IEEE.

[80] Hehn, J., Mendez, D., Uebernickel, F., Brenner, W. and Broy, M., 2019. On integrating design thinking for human-centered requirements engineering. *IEEE Software*, *37*(2), pp.25-31.

[81] Higuchi, M.M. and Nakano, D.N., 2017. Agile design: A combined model based on design thinking and agile methodologies for digital games projects. *Revista de Gestão e Projetos*, *8*(2), pp.109-126.

[82] Hildenbrand, T. and Meyer, J., 2012. Intertwining lean and design thinking: software product development from empathy to shipment. *Software for people: Fundamentals, trends and best practices*, pp.217-237.

[83] Hiremath, M. and Sathiyam, V., 2013. Fast train to DT: a practical guide to coach design thinking in software industry. In *Human-Computer Interaction–INTERACT 2013: 14th IFIP TC 13 International Conference, Cape Town, South Africa, September 2-6, 2013, Proceedings, Part III 14* (pp. 780-787). Springer Berlin Heidelberg.

[84] Hoffman Libby, 2016. 10 Models for Design Thinking. *Medium*.

[85] Hossain, E., Bannerman, P.L. and Jeffery, D.R., 2011, June. Scrum practices in global software development: a research framework. In *International conference on product focused software process improvement* (pp. 88-102). Berlin, Heidelberg: Springer Berlin Heidelberg.

[86] Hussain, Z., Lechner, M., Milchrahm, H., Shahzad, S., Slany, W., Umgeher, M. and Wolkerstorfer, P., 2008. Agile user-centered design applied to a mobile multimedia streaming application. In *HCI and Usability for Education and Work: 4th Symposium of the Workgroup Human-Computer Interaction and Usability Engineering of the Austrian Computer Society, USAB 2008, Graz, Austria, November 20-21, 2008. Proceedings 4* (pp. 313-330). Springer Berlin Heidelberg.

[87] Iamratanakul, S., Shankar, R. and Dimmitt, N.J., 2009, August. Improving project portfolio management with strategic alignment. In *PICMET'09-2009 Portland International Conference on Management of Engineering & Technology* (pp. 1290-1300). IEEE.

[88] Inayat, I., Salim, S.S., Marczak, S., Daneva, M. and Shamshirband, S., 2015. A systematic literature review on agile requirements engineering practices and challenges. *Computers in human behavior*, *51*, pp.915-929.

[89] Isa, W.A.R.W.M., Lokman, A.M., Aris, S.R.S., Aziz, M.A., Taslim, J., Manaf, M. and Sulaiman, R., 2014, May. Engineering rural informatics using agile user

centered design. In *2014 2nd International Conference on Information and Communication Technology (ICoICT)* (pp. 367-372). IEEE.
[90] ISO Standard AC08206635, A. ed., 2010. *Ergonomics of human-system interaction-Part 210: Human-centred design for interactive systems (ISO 9241-210: 2010)*. ISO.
[91] Israel, G.D., 1992. *Determining Sample Size.* Gainesville: University of Florida Cooperative Extension Service, Institute of Food and Agricultural Sciences.
[92] Jarke, M., Loucopoulos, P., Lyytinen, K., Mylopoulos, J. and Robinson, W., 2011. The brave new world of design requirements. *Information Systems*, *36*(7), pp.992-1008.
[93] Joshi, A., Kale, S., Chandel, S. and Pal, D.K., 2015. Likert scale: Explored and explained. *British Journal of Applied Science & Technology*, 7(4), pp.396–403. doi:10.9734/BJAST/2015/14975
[94] Kalenda, M., Hyna, P. and Rossi, B., 2018. Scaling agile in large organizations: Practices, challenges, and success factors. *Journal of Software: Evolution and Process*, *30*(10), p.e1954.
[95] Keller, K.L. and Kotler, P., 2009. *Marketing management* (p. 816). Hoboken: Pearson Prentice Hall.
[96] Kerzner, H., 2017. *Project management metrics, KPIs, and dashboards: A guide to measuring and monitoring project performance*. 3rd ed. Hoboken, NJ: John Wiley & Sons.
[97] Kettunen, P. and Laanti, M., 2008. Combining agile software projects and large-scale organizational agility. Software Process: Improvement and Practice, 13(2), pp.183-193. https://doi.org/10.1002/spip.371
[98] Knaster, R. and Leffingwell, D., 2020. *SAFe 5.0 distilled: achieving business agility with the scaled agile framework*. Addison-Wesley Professional.
[99] Lárusdóttir, M., Cajander, Å. and Gulliksen, J., 2014. Informal feedback rather than performance measurements – user-centred evaluation in Scrum projects. *Behaviour & Information Technology*, *33*(11), pp.1118-1135.
[100] Lester, C.Y., 2011. Combining agile methods and user-centered design to create a unique user experience: An empirical inquiry. In *The Fourth International Conference on Advances in Computer-Human Interactions, ACHI 2011*.
[101] Likert, R., 1932. A technique for the measurement of attitudes. *Archives of Psychology*, 22(140), pp.1–55.
[102] Lindberg, T., Meinel, C. and Wagner, R., 2011. Design thinking: A fruitful concept for IT development?. *Design thinking: Understand–improve–apply*, pp.3-18.

[103] Losada, B., Urretavizcaya, M. and de Castro, I.F., 2011. An integrated approach to develop interactive software. In *Human-Computer Interaction–INTERACT 2011: 13th IFIP TC 13 International Conference, Lisbon, Portugal, September 5-9, 2011, Proceedings, Part IV 13* (pp. 470-474). Springer Berlin Heidelberg.

[104] Lucena, P., Braz, A., Chicoria, A. and Tizzei, L., 2017. IBM design thinking software development framework. In *Agile Methods: 7th Brazilian Workshop, WBMA 2016, Curitiba, Brazil, November 7-9, 2016, Revised Selected Papers 7* (pp. 98-109). Springer International Publishing.

[105] Luedeke, T.F., Köhler, C., Conrad, J., Grashiller, M., Sailer, A. and Vielhaber, M., 2018. CPM/PDD in the context of design thinking and agile development of cyber-physical systems. *DS 91: Proceedings of NordDesign 2018, Linköping, Sweden, 14th-17th August 2018*.

[106] Lukasik, R. and Saylor, J., 2018. Agile, meet design thinking - Get better experiences to market faster. *IBM Institute for Business Value*, *3*.

[107] Magare, A., Lamin, M. and Chakrabarti, P., 2021. Inherent mapping analysis of agile development methodology through design thinking. In *Data Science and Intelligent Applications: Proceedings of ICDSIA 2020* (pp. 527-534). Springer Singapore.

[108] Marion, T., Cannon, D., Reid, T. and McGowan, A.M., 2021. A conceptual model for integrating design thinking and lean startup methods into the innovation process. *Proceedings of the Design Society*, *1*, pp.31-40.

[109] Mayhew, D.J., 1999, May. The usability engineering lifecycle. In *CHI'99 Extended Abstracts on Human Factors in Computing Systems* (pp. 147-148).

[110] Menchaca, R., Donnellan, N., Wintrich, G. and Donnellan, B., 2014. Applying Design Thinking throughout the Product Lifecycle in Dell Inc. In *Design Science: Perspectives from Europe: European Design Science Symposium, EDSS 2013, Dublin, Ireland, November 21-22, 2013. Revised Selected Papers* (pp. 75-87). Springer International Publishing.

[111] Mendonça de Sá Araújo, C.M., Miranda Santos, I., Dias Canedo, E. and Favacho de Araújo, A.P., 2019. Design thinking versus design sprint: A comparative study. In *Design, User Experience, and Usability. Design Philosophy and Theory: 8th International Conference, DUXU 2019, Held as Part of the 21st HCI International Conference, HCII 2019, Orlando, FL, USA, July 26–31, 2019, Proceedings, Part I 21* (pp. 291-306). Springer International Publishing.

[112] Microsoft (2024) *QUARTILE.INC function*. Available at: https://support.microsoft.com/en-us/office/quartile-inc-function (Accessed: 22 August 2025).
[113] Mohd Arof, Khairul & Ismail, Syuhaida & Saleh, Abd Latif. (2018). Contractor's Performance Appraisal System in the Malaysian Construction Industry: Current Practice, Perception and Understanding. International Journal of Engineering & Technology. 7. 46. 10.14419/ijet.v7i3.9.15272.
[114] Mohd Arof, Khairul & Ismail, Syuhaida & Saleh, Abd Latif. (2018). Contractor's Performance Appraisal System in the Malaysian Construction Industry: Current Practice, Perception and Understanding. International Journal of Engineering & Technology. 7. 46. 10.14419/ijet.v7i3.9.15272.
[115] Molamphy, D., Fitzgerald, B. and Conboy, K., 2025, April. Customer Validation, Feedback and Collaboration in Large-Scale Continuous Software Development. In *2025 IEEE/ACM 47th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)* (pp. 178-180). IEEE.
[116] Molokken, K. and Jorgensen, M., 2003, September. A review of software surveys on software effort estimation. In *2003 International Symposium on Empirical Software Engineering, 2003. ISESE 2003. Proceedings.* (pp. 223-230). IEEE.
[117] Moralles, C., Zorzetti, M., Signoretti, I., Pereira, E., Vaccaro, M., Prauchner, B., Salerno, L., Trindade, C., Marczak, S. and Bastos, R., 2020. On the development of a model to support the combined use of agile software development with user-centered design and lean startup. In *Systems, Software and Services Process Improvement: 27th European Conference, EuroSPI 2020, Düsseldorf, Germany, September 9–11, 2020, Proceedings 27* (pp. 220-231). Springer International Publishing.
[118] Morris, P.W., 2013. Reconstructing project management. John Wiley & Sons.
[119] Müller, R., 2016. Organizational project governance. In *Governance and Governmentality for Projects* (pp. 25-38). Routledge.
[120] Nedeltcheva, G.N. and Shoikova, E., 2017, December. Coupling design thinking, user experience design and agile: towards cooperation framework. In *Proceedings of the international conference on big data and internet of thing* (pp. 225-229).
[121] Newman, P., Ferrario, M.A., Simm, W., Forshaw, S., Friday, A. and Whittle, J., 2015, May. The role of design thinking and physical prototyping in social software engineering. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering* (Vol. 2, pp. 487-496). IEEE.

[122] Nielsen, J., 1994. *Usability engineering*. Morgan Kaufmann.
[123] Nielsen, J., 1995. Applying discount usability engineering. *IEEE software*, *12*(1), pp.98-100.
[124] Oktay, J.S., 2012. *Grounded theory*. Pocket Guide to Social Work Re.
[125] Paasivaara, M., Behm, B., Lassenius, C. and Hallikainen, M., 2018. Large-scale agile transformation at Ericsson: a case study. *Empirical Software Engineering*, *23*(5), pp.2550-2596.
[126] Pai, A.R., Joshi, G. and Rane, S., 2021. Quality and reliability studies in software defect management: a literature review. *International Journal of Quality & Reliability Management*, *38*(10), pp.2007-2033.
[127] Parizi, R., da Silva, M.M., Couto, I., Trindade, K., Prestes, M.P., dos Santos Marczak, S., Conte, T. and Candello, H., 2020. Design thinking in software requirements: What techniques to use? A Proposal for a Recommendation Tool. In *Proceedings of the XXIII Ibero-American Conference on Software Engineering-CIbSE 2020, 2020, Estados Unidos.*
[128] Parizi, R., Prestes, M., Marczak, S. and Conte, T., 2022. How has design thinking being used and integrated into software development activities? A systematic mapping. *Journal of Systems and Software*, *187*, p.111217.
[129] Pereira, J.C. and de FSM Russo, R., 2018. Design thinking integrated in agile software development: A systematic literature review. *Procedia computer science*, *138*, pp.775-782.
[130] Pereira, L., Parizi, R., Prestes, M., Marczak, S. and Conte, T., 2021, March. Towards an understanding of benefits and challenges in the use of design thinking in requirements engineering. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing* (pp. 1338-1345).
[131] Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M. (2008). Systematic mapping studies in software engineering. Proceedings of the International Conference on Evaluation and Assessment in Software Engineering, ACM, Bari, Italy, pp. 1-10.
[132] Pikkarainen, M., Haikara, J., Salo, O., Abrahamsson, P. and Still, J., 2008. The impact of agile practices on communication in software development. *Empirical Software Engineering*, *13*(3), pp.303-337.
[133] PMI (Project Management Institute), 2022. A guide to the project management body of knowledge (PMBOK® guide). 7th ed. Newtown Square, PA: Project Management Institute.
[134] PMI, 2017. *The Standard for Program Management*. 4th ed. Newtown Square, PA: Project Management Institute.

[135] Podsakoff, P.M. et al., 2003. Common method biases in behavioral research: A critical review. *Journal of Applied Psychology*, 88(5), pp.879–903.

[136] Prasad, W.R., Perera, G.I.U.S., Padmini, K.J. and Bandara, H.D., 2018, May. Adopting design thinking practices to satisfy customer expectations in agile practices: a case from Sri Lankan software development industry. In *2018 Moratuwa Engineering Research Conference (MERCon)* (pp. 471-476). IEEE.

[137] Prior, S., Waller, A., Black, R. and Kroll, T., 2013, April. Use of an agile bridge in the development of assistive technology. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 1579-1588).

[138] Project Management Institute (PMI), 2023. *Global Project Management Talent Gap*. [online] Project Management Institute. Available at: https://www.pmi.org/learning/thought-leadership/global-project-management-talent-gap [Accessed 15 July 2025].

[139] Przybilla, L., Schreieck, M., Klinker, K., Pflügler, C., Wiesche, M. and Krcmar, H., 2018. Combining design thinking and agile development to master highly innovative IT projects.

[140] Ramesh, B., Cao, L. and Baskerville, R., 2010. Agile requirements engineering practices and challenges: an empirical study. *Information Systems Journal*, *20*(5), pp.449-480.

[141] Ries, E., 2011. *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. New York: Crown Business.

[142] Rigby, D.K., Sutherland, J. and Takeuchi, H., 2016. The secret history of agile innovation. *Harvard Business Review*, *4*.

[143] Rindfleisch, A. et al., 2008. Cross-sectional versus longitudinal survey research. *Journal of Marketing Research*, 45(3), pp.261–279.

[144] Sandino, D., Matey, L.M. and Vélez, G., 2013. Design thinking methodology for the design of interactive real-time applications. In *Design, User Experience, and Usability. Design Philosophy, Methods, and Tools: Second International Conference, DUXU 2013, Held as Part of HCI International 2013, Las Vegas, NV, USA, July 21-26, 2013, Proceedings, Part I 2* (pp. 583-592). Springer Berlin Heidelberg.

[145] Sawhney, M., Verona, G. and Prandelli, E., 2005. Collaborating to create: The Internet as a platform for customer engagement in product innovation. *Journal of interactive marketing*, *19*(4), pp.4-17.

[146] Schön, E.M., Winter, D., Uhlenbrok, J., Escalona Cuaresma, M.J. and Thomaschewski, J., 2016. Enterprise experience into the integration of human-centered design and Kanban. In *ICSOFT-EA 2016: 11th International Joint*

Conference on Software Technologies (2016), p 133-140. ScitePress Digital Library.

[147] Schröer, C., Kruse, F. and Gómez, J.M., 2021. A systematic literature review on applying CRISP-DM process model. *Procedia Computer Science*, *181*, pp.526-534.

[148] Schwaber, K. and Sutherland, J., 2020. The Scrum Guide. Available at: https://scrumguides.org (Accessed: 26 August 2025).

[149] Sebok, A., Walters, B. and Plott, C., 2017, September. Integrating human-centered design and the agile development process for safety and mission critical system development. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (Vol. 61, No. 1, pp. 1086-1090). Sage CA: Los Angeles, CA: SAGE Publications.

[150] Serrador, P. and Pinto, J.K., 2015. Does Agile work? — A quantitative analysis of agile project success. International journal of project management, 33(5), pp.1040-1051.

[151] Signoretti, I., Salerno, L., Marczak, S. and Bastos, R., 2020. Combining user-centered design and lean startup with agile software development: a case study of two agile teams. In *Agile Processes in Software Engineering and Extreme Programming: 21st International Conference on Agile Software Development, XP 2020, Copenhagen, Denmark, June 8–12, 2020, Proceedings 21* (pp. 39-55). Springer International Publishing.

[152] Silva, N., Cunha, J.C. and Vieira, M., 2017. A field study on root cause analysis of defects in space software. *Reliability Engineering & System Safety*, *158*, pp.213-229.

[153] Singla, K., Bose, J. and Naik, C., 2018, December. Analysis of software engineering for agile machine learning projects. In *2018 15th IEEE India Council International Conference (INDICON)* (pp. 1-5). IEEE.

[154] Snyder, H., 2019. Literature review as a research methodology: An overview and guidelines. *Journal of business research*, *104*, pp.333-339.

[155] Sohaib, O. and Khan, K., 2011. Incorporating discount usability in extreme programming. *International journal of software engineering and its applications*, *5*(1), pp.51-62.

[156] Sohaib, O., Solanki, H., Dhaliwa, N., Hussain, W. and Asif, M., 2019. Integrating design thinking into extreme programming. *Journal of Ambient Intelligence and Humanized Computing*, *10*, pp.2485-2492.

[157] Steinke, G., Al-Deen, M. and LaBrie, R., 2017, March. Innovating information system development methodologies with design thinking. In *Proceedings of*

International Conference on Applied Innovation in IT (Vol. 5, No. 1, pp. 51-55). Anhalt University of Applied Sciences.
[158] Strauss, A. and Corbin, J.M., 1997. *Grounded theory in practice*. Sage.
[159] Stray, V., Moe, N.B. and Hoda, R., 2018, May. Autonomous agile teams: challenges and future directions for research. In *Proceedings of the 19th international conference on agile software development: companion* (pp. 1-5).
[160] Sue, V.M. and Ritter, L.A., 2012. *Conducting Online Surveys*. 2nd ed. Thousand Oaks: Sage Publications.
[161] Tellioğlu, H., 2016. Models as bridges from design thinking to engineering. In *Proceedings of the 10th International Conference on Interfaces and Human Computer Interaction (IHCI 2016), Multi Conference on Computer Science and Information Systems, July* (pp. 1-4).
[162] Tobak, S.: 9 Reasons why most startups fail (2014). http://www.entrepreneur.com/article/ 231129
[163] Torraco, R.J., 2005. Writing integrative literature reviews: Guidelines and examples. *Human resource development review*, *4*(3), pp.356-367.
[164] TRIVISIOS, A.N., 1987. Introduction to social science research: Qualitative research in education. *A pesquisa*, p.133.
[165] Tukey, J.W. (1977) *Exploratory Data Analysis*. Reading, MA: Addison-Wesley.
[166] Unterkalmsteiner, M., Gorschek, T., et al., 2023. Evaluation and Measurement of Software Process Improvement – A Systematic Literature Review. *arXiv preprint* arXiv:2307.13143.
[167] Vilkki, K. (2010) 'When agile is not enough', in Lean Enterprise Software and Systems, Springer, 44–47.
[168] Walch, K., 2020. Why Agile Methodologies Miss The Mark For AI & ML Projects. *Retrieved*, *4*(17), p.2021.
[169] Walker, D. and Myrick, F., 2006. Grounded theory: An exploration of process and procedure. *Qualitative health research*, *16*(4), pp.547-559.
[170] Weber, C., CPM/PDD–An Extended Theoretical Approach to Modelling Products and Product Development Processes, 2nd German-Israeli Symposium on Advances in Methods and Systems for Development of Products and Processes, Berlin 07–08.07. 2005. *Bley, H.; Jansen, H.; Krause, F.-L*, pp.159-179.
[171] Wilmshurst, D. and Quick, L., 2023. SAFe® Coaches Handbook: Proven tips and techniques for launching and running SAFe® Teams, ARTs, and Portfolios in an Agile Enterprise. Packt Publishing Ltd.

[172] Wölbling, A., Krämer, K., Buss, C.N., Dribbisch, K., LoBue, P. and Taherivand, A., 2012. Design thinking: An innovative concept for developing user-centered software. *Software for people: Fundamentals, trends and best practices*, pp.121-136.
[173] Wright, K.B., 2005. Researching internet-based populations: Advantages and disadvantages of online survey research, online questionnaire authoring software packages, and web survey services. *Journal of Computer-Mediated Communication*, 10(3).
[174] Ximenes, B.H., Alves, I.N. and Araújo, C.C., 2015. Software project management combining agile, lean startup and design thinking. In *Design, User Experience, and Usability: Design Discourse: 4th International Conference, DUXU 2015, Held as Part of HCI International 2015, Los Angeles, CA, USA, August 2–7, 2015, Proceedings, Part I* (pp. 356-367). Springer International Publishing.
[175] Xiong, Y. and Wang, A., 2010, December. A new combined method for UCD and software development and case study. In *The 2nd International Conference on Information Science and Engineering* (pp. 1-4). IEEE.
[176] Xu, X., Lu, Y., Vogel-Heuser, B. and Wang, L., 2021. Industry 4.0 and Industry 5.0—Inception, conception and perception. *Journal of Manufacturing Systems*, *61*, pp.530-535.
[177] Zorzetti, M., Moralles, C., Salerno, L., Pereira, E., Marczak, S. and Bastos, R., 2021. Adopting Agile software development combined with user-centered design and lean startup: a systematic literature review on maturity models. In *International Conference on Enterprise Information Systems* (pp. 517-541). Springer, Cham.
[178] Zorzetti, M., Signoretti, I., Pereira, E., Salerno, L., Moralles, C., Trindade, C., Machado, M., Bastos, R. and Marczak, S., 2020. A practice-informed conceptual model for a combined approach of agile, user-centered design, and lean startup. In *Product-Focused Software Process Improvement: 21st International Conference, PROFES 2020, Turin, Italy, November 25–27, 2020, Proceedings 21* (pp. 142-150). Springer International Publishing.
[179] Zorzetti, M., Signoretti, I., Salerno, L., Marczak, S. and Bastos, R., 2022. Improving agile software development using user-centered design and lean startup. *Information and Software Technology*, *141*, p.106718.

APPENDIX A

SURVEY COVER LETTER

Why some projects succeed spectacularly while others fail?

I am currently a DBA student at Swiss School of Business and Administration, Geneva. I am exploring this question for my Doctorate research & would love your quick input please - just a few minutes, fully confidential. This survey is completely voluntary and we are not capturing any personal data.

Your insights & experiences could be the key to uncover the answer. Once you have filled it in, could you please forward the link to your friends & colleagues.

Thank you for taking the time to share your valuable experiences.

Thanks & Regards,

Amol Kate

APPENDIX B

PYTHON PROGRAMM FOR DATA ANALYSIS & MODEL BUILDING

We have built a Python program for data analysis, building various statistical, ANOVA, Regression and Machine Learning Models. Python script is coded and run in Jupyter Notebook which is an open-source web application with Anaconda ecosystem environment.

Following is the complete Python code, which was used for producing results, as described in earlier chapter:

Import Packages and Libraries:

```
#Install required modules and packages:
!pip install missingno
!pip install pingouin
#Import libraries:
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import missingno as msno
from scipy import stats
import statsmodels.api as sm
import statsmodels.formula.api as smf
from pingouin import cronbach_alpha
import warnings
warnings.filterwarnings( "ignore")
```

Exploratory Data Analysis (EDA)

```
#Read the data from excel file as a dataframe
df=pd.read_excel('Survey_Data.xlsx')
```

```
# To check shape of the data
df.shape
```

```
(302, 21)
```

```
# Check top 5 record in the dataset
df.head()
```

```
# Check bottom 5 record in the dataset
df.tail()
```

```
# Check information of the dataframe
# To check what type data is there in each column (Data type)
df.info()
```

```
# Make different list for categorical columns and numerical columns
cat=[]
num=[]
for i in df.columns:
    if df[i].dtype=="object":
        cat.append(i)
    else:
        num.append(i)
print('Number of Categorical columns =')
print(cat)

print('\n And Number of Numerical columns =')
print(num)
print('\n')
```

```
# Find categorical values in each of the object type columns (in Industry & in Method):
for column in df.columns:
    if df[column].dtype == 'object':
        print(column.upper(),': ',df[column].nunique())
        print(df[column].value_counts(normalize = True).sort_values())
        print('\n')
```

```
# Check for any null values in the dataset:
df.isnull().sum()
```

```
# Check for any duplicate data records in the dataset:
dups = df.duplicated()
print('Number of duplicate rows = %d' % (dups.sum()))

df[dups]
```

```
# Check summary of the dataframe:
df_stat= df.describe().T
df_stat['Range']= df_stat['max'] - df_stat['min']
df_stat['IQR']= df_stat['75%'] - df_stat['25%']
df_stat['Variance']= df_stat['std']*df_stat['std']
df_stat['Coeff of Variation']= df_stat['std']/df_stat['mean']
df_stat

# Drop the 'count' column
df_stat = df_stat.drop('count', axis=1)

df_stat
```

```python
# Create the Boxplot
plt.figure(figsize=(40, 20))
# Changed to horizontal orientation of bars
boxplot = sns.boxplot(data=df, orient='h', palette='Set2')
boxplot.axes.set_title('BoxPlot', fontsize=35)
# Swapped x & y labels for horizontal orientation
boxplot.set_xlabel('Values', fontsize=35)
boxplot.set_ylabel('Parameters', fontsize=35)
boxplot.tick_params(labelsize=35)

# Function to identify outliers for each column
def find_outliers(dataframe):
    outlier_info = {}

    for column in df.select_dtypes(include=[np.number]).columns:
        # Calculate Q1, Q3, and IQR
        Q1 = df[column].quantile(0.25)
        Q3 = df[column].quantile(0.75)
        IQR = Q3 - Q1

        # Define outlier boundaries
        lower_bound = Q1 - 1.5 * IQR
        upper_bound = Q3 + 1.5 * IQR
        # Find outliers
        outliers = df[(df[column] < lower_bound) | (df[column] > upper_bound)][column]

        if not outliers.empty:
            outlier_info[column] = {
                'count': len(outliers),
                'percentage': (len(outliers) / len(df)) * 100,
                'min_outlier': outliers.min() if len(outliers) > 0 else None,
                'max_outlier': outliers.max() if len(outliers) > 0 else None,
                'boundaries': (lower_bound, upper_bound)
            }

    return outlier_info

# Get outlier information
outliers = find_outliers(df)

# Print outlier information
print("Outlier Analysis:")
for column, info in outliers.items():
    print(f"\n{column}:")
    print(f"  Number of outliers: {info['count']}")
    print(f"  Percentage of data: {info['percentage']:.2f}%")
    print(f"  Outlier range: {info['min_outlier']} to {info['max_outlier']}")
    print(f"  Normal range should be: {info['boundaries'][0]:.2f} to {info['boundaries'][1]:.2f}")

plt.tight_layout()
plt.show()
```

```python
# Draw Pairplots for all Variables

from matplotlib.colors import LinearSegmentedColormap

# Set an aesthetic style and color palette
sns.set_style("ticks")
plt.rcParams['figure.figsize'] = [20, 20]

# Create a custom colormap for various colors
colors = ["#4B0082", "#0000FF", "#00FFFF", "#00FF00", "#FFFF00", "#FF7F00", "#FF0000"]
custom_cmap = LinearSegmentedColormap.from_list("custom_colormap", colors)

# Create a pairplot
pairplot = sns.pairplot(
    df,
    diag_kind='kde',
    corner=True,  # Only show the lower triangle to reduce redundancy
    plot_kws={
        'alpha': 0.6,  # Add transparency to see overlapping points
        'edgecolor': 'none',
        's': 30,  # Adjust point size
    },
    diag_kws={
        'fill': True,
        'alpha': 0.6,
        'linewidth': 1.5,
        'shade': True,
        'color': 'steelblue'

    },
    palette='viridis',  # Use a colorful palette (Notes: plasma', 'magma', 'cividis')
    height=2.5  # Adjust subplot size
)

# Customize the appearance
pairplot.fig.suptitle('Pairplot Visualization', y=1.02, fontsize=40)

# Improve the layout
plt.tight_layout()

plt.show()
```

```python
# Spearman Correlations

# Select only numeric columns
numeric_df = df.select_dtypes(include=['number'])
df_corr = numeric_df.corr()
df_corr
```

```python
# Spearman Correlation Matrix Heatmap:

#Set up the matplotlib figure
plt.figure(figsize=(12, 10))

# Create a mask for the upper triangle to avoid redundancy
mask = np.triu(np.ones_like(df_corr, dtype=bool))
```

```python
# Colormap option: 'magma' - perceptually uniform (black to yellow)
cmap = 'magma'

# Draw the heatmap with the mask and correct aspect ratio
sns.heatmap(df_corr,
            mask=mask,
            cmap=cmap,
            vmax=1,
            vmin=-1,
            center=0,
            square=True,
            linewidths=.5,
            cbar_kws={"shrink": .5},
            annot=True,  # Adds correlation values to cells
            fmt=".2f")   # Format for the annotations (2 decimal places)

# Adjust the plot
plt.tight_layout()
plt.title('Spearman Correlation Matrix Heatmap', fontsize=16, pad=20)

# Show the plot
plt.show()
```

Principal Component Analysis (PCA)

```python
# Principal Component Analysis (PCA):

from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

# Set the style for more beautiful plots
sns.set(style="whitegrid")
plt.rcParams['font.family'] = 'sans-serif'
plt.rcParams['font.sans-serif'] = ['Arial']
plt.rcParams['axes.edgecolor'] = '#333333'
plt.rcParams['axes.linewidth'] = 0.8
plt.rcParams['xtick.color'] = '#333333'
plt.rcParams['ytick.color'] = '#333333'

# Select only dependent variables
dependent_vars = [
    'y1_Schedule',
    'y2_Cost',
    'y3_Quality',
    'y4_Scope',
    'y5_Customer'
]
X = df[dependent_vars]

# Standardize the data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

```python
# Apply PCA
pca = PCA(n_components=5)
principal_components = pca.fit_transform(X_scaled)

# Create a DataFrame for principal components
pc_df = pd.DataFrame(data=principal_components,
                     columns=['PC1', 'PC2', 'PC3', 'PC4', 'PC5'])

# Add PC1 back to original dataset as composite score
df['Y_Project_Success'] = pc_df['PC1']

# Display PCA Loadings (components)
loadings = pd.DataFrame(pca.components_.T,
                        index=dependent_vars,
                        columns=['PC1', 'PC2', 'PC3', 'PC4', 'PC5'])

print("\nPCA Loadings (Variable Contributions):")
print(loadings)

# Explained Variance Ratio
explained_variance = pd.DataFrame({
    'Principal Component': ['PC1', 'PC2', 'PC3', 'PC4', 'PC5'],
    'Explained Variance Ratio': pca.explained_variance_ratio_
})
print("\nExplained Variance by Each Component:")
print(explained_variance)

# Heatmap of Loadings
plt.figure(figsize=(10, 6))
sns.heatmap(loadings, annot=True, cmap='coolwarm', center=0)
plt.title('PCA Loadings Heatmap', fontsize=16, fontweight='bold')
plt.tight_layout()
plt.show()
# Enhanced Scree plot
fig, ax = plt.subplots(figsize=(12, 7))

# Calculate cumulative explained variance
cum_explained_variance = np.cumsum(pca.explained_variance_ratio_)

# Create a beautiful scree plot with dual y-axis
bars = ax.bar(
    range(1, 6),
    pca.explained_variance_ratio_,
    alpha=0.7,
    color='#5975A4',
    width=0.5,
    edgecolor='black',
    linewidth=1.5,
    label='Individual Explained Variance'
)

# Add value labels on top of bars
for bar in bars:
    height = bar.get_height()
    ax.text(
        bar.get_x() + bar.get_width()/2.,
```

```python
        height + 0.01,
        f'{height:.2f}',
        ha='center',
        va='bottom',
        fontweight='bold'
    )

# Create a second y-axis for the cumulative line
ax2 = ax.twinx()
ax2.plot(
    range(1, 6),
    cum_explained_variance,
    'o-',
    color='#CC6677',
    linewidth=3,
    markersize=10,
    label='Cumulative Explained Variance'
)

# Add value labels for the line
for i, val in enumerate(cum_explained_variance):
    ax2.text(
        i + 1.1,
        val - 0.03,
        f'{val:.2f}',
        color='#CC6677',
        fontweight='bold'
    )

# Add a horizontal line at 0.8 (80%) for reference
ax2.axhline(y=0.8, color='gray', linestyle='--', alpha=0.7)
ax2.text(4.5, 0.82, '80% Threshold', va='center', alpha=0.7)
# Customize the plot
ax.set_xlabel('Principal Component', fontsize=14, fontweight='bold')
ax.set_ylabel('Individual Explained Variance Ratio', fontsize=14, fontweight='bold')
ax2.set_ylabel('Cumulative Explained Variance Ratio', fontsize=14, fontweight='bold')
ax.set_title('Scree Plot with Cumulative Variance', fontsize=18, fontweight='bold', pad=20)
ax.set_xticks(range(1, 6))
ax.set_xticklabels([f'PC{i}' for i in range(1, 6)], fontsize=12)
ax.tick_params(axis='y', labelsize=12)
ax2.tick_params(axis='y', labelsize=12)

# Add a grid but only on the x-axis
ax.grid(axis='y', linestyle='--', alpha=0.7)

# Add legends
lines1, labels1 = ax.get_legend_handles_labels()
lines2, labels2 = ax2.get_legend_handles_labels()
ax.legend(lines1 + lines2, labels1 + labels2, loc='upper left', fontsize=12)

# Add a subtle background color
fig.patch.set_facecolor('#F8F8F8')
ax.set_facecolor('#F8F8F8')

plt.tight_layout()
plt.show()
```

```python
# Download Principal Component Analysis (PCA) File in Excel format:

# Using the 'df' DataFrame that contains the PCA results from your analysis
# This code will download the DataFrame with the Y_Project_Success column in Excel format

# First, make sure the DataFrame is saved to an Excel file
df_with_pca_path = 'data_with_pca.xlsx'
df.to_excel(df_with_pca_path, index=False)  # Changed to_csv to to_excel
print(f"DataFrame with PCA components saved to {df_with_pca_path}")

# Option 1: Download using FileLink (works in Jupyter Notebook)
from IPython.display import FileLink
display(FileLink(df_with_pca_path))
```

EDA post PCA:

```python
# Check the dataframe post PCA:
df.head()
```

```python
df.shape
```

```python
# Check summary of the dataframe:
df_stat= df.describe().T
df_stat['Range']= df_stat['max'] - df_stat['min']
df_stat['IQR']= df_stat['75%'] - df_stat['25%']
df_stat['Variance']= df_stat['std']*df_stat['std']
df_stat['Coeff of Variation']= df_stat['std']/df_stat['mean']
df_stat

# Drop the 'count' column
df_stat = df_stat.drop('count', axis=1)

df_stat
```

```python
# Visualize missing values
msno.matrix(df)
```

```python
import math

# Get only numeric columns for histograms
numeric_columns = df.select_dtypes(include=np.number).columns.tolist()

# Calculate grid dimensions
n_cols = 2
n_rows = math.ceil(len(numeric_columns) / n_cols)

# Create a figure with appropriate size
fig, axes = plt.subplots(n_rows, n_cols, figsize=(5*n_cols, 4*n_rows))
# Flatten the axes array to make indexing easier
axes = axes.flatten()

# Set the style
sns.set_style("whitegrid")

# Create a color palette with enough colors
colors = sns.color_palette("husl", len(numeric_columns))
```

```python
# Plot each variable
for i, column in enumerate(numeric_columns):
    # Use the axes array instead of plt.subplot
    ax = axes[i]

    # Create histogram with KDE
    sns.histplot(
        data=df,
        x=column,
        kde=True,
        color=colors[i % len(colors)],
        alpha=0.7,
        edgecolor='black',
        linewidth=1,  # Edge width
        bins=20,  # Number of bins
        ax=ax  # Use the current axis
    )

    # Add mean line
    mean_val = df[column].mean()
    ax.axvline(mean_val, color='red', linestyle='--', linewidth=2,
              label=f'Mean: {mean_val:.2f}')

    # Add median line
    median_val = df[column].median()
    ax.axvline(median_val, color='green', linestyle='-.', linewidth=2,
              label=f'Median: {median_val:.2f}')

    # Add title and labels
    ax.set_title(f'{column}', fontsize=12)
    ax.set_xlabel('')  # Remove x-label to save space

    # Only show y-label for leftmost plots
    if i % n_cols == 0:
        ax.set_ylabel('Frequency', fontsize=10)
    else:
        ax.set_ylabel('')

    # Add legend
    ax.legend()

# Hide any unused subplots
for j in range(i+1, len(axes)):
    axes[j].set_visible(False)

# Adjust layout
plt.tight_layout()
plt.subplots_adjust(top=0.95)
fig.suptitle('Distribution of All Variables', fontsize=16, fontweight='bold')

# Show the plot
plt.show()
```

```python
# Select only numeric columns -
numeric_df = df.select_dtypes(include=['number'])
df_corr = numeric_df.corr()
df_corr
```

```python
# Spearman Correlation Matrix Heatmap AFTER PCA:

# First, select only numeric columns for correlation analysis
numeric_df = df.select_dtypes(include=['number'])  # This selects only numeric columns
df_corr1 = numeric_df.corr()

# Set up the matplotlib figure
plt.figure(figsize=(12, 10))

# Create a mask for the upper triangle
mask = np.triu(np.ones_like(df_corr1, dtype=bool))

# Colormap: magma' - perceptually uniform (black to yellow)
cmap = 'magma'

# Option 7: Custom palette
#cmap = sns.diverging_palette(240, 10, as_cmap=True)  # Blue to Red

# Draw the heatmap with the mask and correct aspect ratio
sns.heatmap(df_corr1,
            mask=mask,
            cmap=cmap,
            vmax=1,
            vmin=-1,
            center=0,
            square=True,
            linewidths=.5,
            cbar_kws={"shrink": .5},
            annot=True,  # Adds correlation values to cells
            fmt=".2f")   # Format for the annotations (2 decimal places)
# Adjust the plot
plt.tight_layout()
plt.title('Spearman Correlation Matrix Heatmap Post PCA', fontsize=16, pad=20)

# Show the plot
plt.show()
```

Cronbach’s Alpha:

```python
# Reliability Check (Cronbach's Alpha) for xi

# Select all Likert-scale independent variables (x1 to x14)
likert_vars = ['x1_Empathy', 'x2_Automation', 'x3_Vision', 'x4_SustEthics', 'x5_ReqSource',
               'x6_Growth', 'x7_Human', 'x8_Scope', 'x9_Governance', 'x10_Strategy',
               'x11_IssueRes', 'x12_Retrospective', 'x13_Impediments', 'x14_Integration']

# Check internal consistency
alpha, _ = cronbach_alpha(data=df[likert_vars])
print(f"Cronbach's Alpha: {alpha:.3f}")
```

```python
# Reliability Check (Cronbach's Alpha) for yi

# Select all Likert-scale independent variables (x1 to x14)
likert_vars = ['y1_Schedule', 'y2_Cost', 'y3_Quality', 'y4_Scope', 'y5_Customer']
```

```
# Check internal consistency
alpha, _ = cronbach_alpha(data=df[likert_vars])
print(f"Cronbach's Alpha: {alpha:.3f}")
```

Multiple Linear Regression (MLR) Model:

```
# Multiple Linear Regression

# Define X and y
# X = df[[f'x{i}' for i in range(1, 16)]]  # x1 to x15
X = df[['x1_Empathy', 'x2_Automation', 'x3_Vision', 'x4_SustEthics', 'x5_ReqSource', 'x6_Growth',
        'x7_Human', 'x8_Scope', 'x9_Governance', 'x10_Strategy', 'x11_IssueRes', 'x12_Retrospective
        'x13_Impediments', 'x14_Integration']]
X = sm.add_constant(X)                  # Add intercept
Y = df['Y_Project_Success']             # Dependent variable

# Fit model
model = sm.OLS(Y, X).fit()

# Display regression results
print(model.summary())
```

MLR Model Dignostics:

```
import pandas as pd
import numpy as np
import statsmodels.api as sm
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score, mean_squared_error


# Load and prepare your data

X = df[['x1_Empathy', 'x2_Automation', 'x3_Vision', 'x4_SustEthics', 'x5_ReqSource',
         'x6_Growth', 'x7_Human', 'x8_Scope', 'x9_Governance', 'x10_Strategy',
         'x11_IssueRes', 'x12_Retrospective', 'x13_Impediments', 'x14_Integration']]
y = df['Y_Project_Success']

# Add constant for intercept term
X = sm.add_constant(X)


# Split into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


# Fit OLS Model
ols_model = sm.OLS(y_train, X_train).fit()

# Predict on training and test data
y_train_pred = ols_model.predict(X_train)
y_test_pred = ols_model.predict(X_test)
```

```
# Calculate R² and RMSE
r2_train = r2_score(y_train, y_train_pred)
r2_test = r2_score(y_test, y_test_pred)
rmse_train = np.sqrt(mean_squared_error(y_train, y_train_pred))
rmse_test = np.sqrt(mean_squared_error(y_test, y_test_pred))

print("Training R²:", round(r2_train, 3))
print("Testing R²:", round(r2_test, 3))
print("Training RMSE:", round(rmse_train, 3))
print("Testing RMSE:", round(rmse_test, 3))


# Optional: Summary of OLS Model
print(ols_model.summary())
```

```
# Residual analysis: residuals vs fitted values for full model
residuals = model.resid
fitted = model.fittedvalues

# Plot residuals to check for patterns
sns.residplot(x=fitted, y=residuals, lowess=True, line_kws={'color': 'red'})
plt.xlabel("Predicted (Fitted) Values from MLR Model")
plt.ylabel("Residuals (Actual - Predicted)")
plt.title("Residuals vs Fitted Values")
plt.axhline(0, linestyle='--', color='black')
plt.show()
```

Non-Linear Model - GAM Model:

```
# First, install the pygam package
!pip install pygam

# Import necessary libraries
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score, mean_squared_error
import numpy as np
import matplotlib.pyplot as plt
from pygam import LinearGAM, s
from statsmodels.nonparametric.smoothers_lowess import lowess  # Import LOWESS for smooth curve

# X and y are already defined in above code.

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create and fit the GAM model
gam = LinearGAM(
    s(0) + s(1) + s(2) + s(3) + s(4) + s(5) + s(6) +
    s(7) + s(8) + s(9) + s(10) + s(11) + s(12) + s(13)
).fit(X_train.values, y_train.values)


# Evaluate model
y_train_pred = gam.predict(X_train)
y_test_pred = gam.predict(X_test)
```

```python
train_r2 = r2_score(y_train, y_train_pred)
test_r2 = r2_score(y_test, y_test_pred)
test_rmse = np.sqrt(mean_squared_error(y_test, y_test_pred))
train_rmse = np.sqrt(mean_squared_error(y_train, y_train_pred))
print("Training R²:", round(train_r2, 3))
print("Testing R²:", round(test_r2, 3))
print("Train RMSE:", round(train_rmse, 3))
print("Test RMSE:", round(test_rmse, 3))
# Plot partial dependence for each predictor
fig, axs = plt.subplots(4, 4, figsize=(16, 12))
for i, ax in enumerate(axs.flatten()[:14]):
    XX = gam.generate_X_grid(term=i)  # Generate grid for feature i

    # Fix: Handle the return values more flexibly
    partial_result = gam.partial_dependence(term=i, X=XX)

    # Check what type of result we got and handle accordingly
    if isinstance(partial_result, tuple):
        # If it's a tuple, extract the first two elements (pdep and confidence intervals)
        pdep = partial_result[0]
        confi = partial_result[1]
    else:
        # If it's not a tuple (e.g., just an array), use it directly
        pdep = partial_result
        confi = None

    # Plot the partial dependence
    ax.plot(XX[:, i], pdep)
    if confi is not None:
        ax.fill_between(XX[:, i], confi[:, 0], confi[:, 1], alpha=0.2)
    ax.set_title(X.columns[i])
plt.tight_layout()
plt.show()
# Plot residuals vs fitted values with LOESS curve
residuals = y_test - y_test_pred
plt.figure(figsize=(8, 6))
plt.scatter(y_test_pred, residuals, color='blue', alpha=0.6)
plt.axhline(0, color='black', linestyle='--')

# Add LOESS smooth curve (red)
# frac parameter controls smoothness (0.5-0.7 is often a good choice)
smoothed = lowess(residuals, y_test_pred, frac=0.6)
plt.plot(smoothed[:, 0], smoothed[:, 1], color='red', linewidth=2)

plt.xlabel("Predicted (Fitted) Values")
plt.ylabel("Residuals (Actual - Predicted)")
plt.title("Residuals vs Fitted Values (GAM Model)")
plt.show()
```

MLR with OLS and all 5 dependent variables Separately:

```
# Multiple Linear Regression considering OLS and all 5 dependent variables:

# Define X and y
# X = df[[f'x{i}' for i in range(1, 16)]]  # x1 to x15
X = df[['x1_Empathy', 'x2_Automation', 'x3_Vision', 'x4_SustEthics', 'x5_ReqSource', 'x6_Growth',
        'x7_Human', 'x8_Scope', 'x9_Governance', 'x10_Strategy', 'x11_IssueRes',
        'x12_Retrospective', 'x13_Impediments', 'x14_Integration']]
X = sm.add_constant(X)                  # Add intercept

# You need to run separate regressions for each dependent variable
# Here's how to do it for each column in Y
dependent_vars = ['y1_Schedule', 'y2_Cost', 'y3_Quality', 'y4_Scope', 'y5_Customer']

# Loop through each dependent variable and fit a separate model
for dep_var in dependent_vars:
    Y = df[dep_var]          # Select a single dependent variable
    # Fit model
    model = sm.OLS(Y, X).fit()
    # Display regression results
    print(f"\nRegression results for {dep_var}:")
    print(model.summary())
```

MANOVA (Multivariate Analysis of Variance)

```
# Multivariate Regression Techniques - MANOVA (Multivariate Analysis of Variance)
# Multiple Linear Regression with multiple dependent variables
import statsmodels.api as sm
import numpy as np
from statsmodels.multivariate.manova import MANOVA

# Define X and y
X = df[['x1_Empathy', 'x2_Automation', 'x3_Vision', 'x4_SustEthics', 'x5_ReqSource',
        'x6_Growth', 'x7_Human', 'x8_Scope', 'x9_Governance', 'x10_Strategy',
        'x11_IssueRes', 'x12_Retrospective', 'x13_Impediments', 'x14_Integration']]
Y = df[['y1_Schedule', 'y2_Cost', 'y3_Quality', 'y4_Scope', 'y5_Customer']]

# Add constant to X for intercept
X = sm.add_constant(X)

# Fit MANOVA model
manova = MANOVA(Y.values, X.values)
print(manova.mv_test())

# If you want individual regression results for each dependent variable
# but want to present them together:
results = {}
for dep_var in Y.columns:
    model = sm.OLS(df[dep_var], X).fit()
    results[dep_var] = model

# Print a summary of all models
print("\nSummary of all regression models:")
for dep_var, model in results.items():
    print(f"\n=== Results for {dep_var} ===")
    print(f"R-squared: {model.rsquared:.4f}")
```

```python
    print(f"Adjusted R-squared: {model.rsquared_adj:.4f}")
    print(f"F-statistic: {model.fvalue:.4f}")
    print(f"Prob (F-statistic): {model.f_pvalue:.4f}")

    # Print coefficients and p-values
    print("\nCoefficients:")
    for i, var in enumerate(X.columns):
        coef = model.params[i]
        p_val = model.pvalues[i]
        print(f"{var}: {coef:.4f} (p={p_val:.4f})")
```

```python
# Coefficient heatmap: Predictor vs project success Outcome
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np
import statsmodels.api as sm
from matplotlib.colors import LinearSegmentedColormap
# Set global font size
plt.rcParams.update({'font.size': 14})

# Define X and y
X = df[['x1_Empathy', 'x2_Automation', 'x3_Vision', 'x4_SustEthics', 'x5_ReqSource',
        'x6_Growth', 'x7_Human', 'x8_Scope', 'x9_Governance', 'x10_Strategy',
        'x11_IssueRes', 'x12_Retrospective', 'x13_Impediments', 'x14_Integration']]
Y = df[['y1_Schedule', 'y2_Cost', 'y3_Quality', 'y4_Scope', 'y5_Customer']]

# Add constant to X for intercept
X = sm.add_constant(X)

# Create a DataFrame to store coefficients
coef_df = pd.DataFrame(index=X.columns)

# Fit models and extract coefficients
for dep_var in Y.columns:
    model = sm.OLS(df[dep_var], X).fit()
    coef_df[dep_var] = model.params

# Remove the constant row for better visualization
coef_df = coef_df.drop('const')

# Create a custom colormap: orange-yellow-green (inverted from previous)
colors = ['orange', 'yellow', 'green']
custom_cmap = LinearSegmentedColormap.from_list('OrangeYellowGreen', colors)

# Create a heatmap with the custom colormap
plt.figure(figsize=(14, 12))
heatmap = sns.heatmap(coef_df, annot=True, cmap=custom_cmap,
                      fmt='.3f', linewidths=.5, annot_kws={"size": 16})
plt.title('Coefficient Heatmap: Predictors vs Project Success Outcomes', fontsize=24)
plt.xlabel('Project Success Outcomes', fontsize=22)
plt.ylabel('Predictors', fontsize=22)
plt.xticks(fontsize=18)
plt.yticks(fontsize=18)
```

```python
# Add a colorbar label
cbar = heatmap.collections[0].colorbar
cbar.set_label('Coefficient Value', rotation=270, labelpad=30, fontsize=20)
cbar.ax.tick_params(labelsize=16)

# Create a second heatmap with p-values to show significance
plt.figure(figsize=(14, 12))
pval_df = pd.DataFrame(index=X.columns)

# Fit models and extract p-values
for dep_var in Y.columns:
    model = sm.OLS(df[dep_var], X).fit()
    pval_df[dep_var] = model.pvalues

# Remove the constant row
pval_df = pval_df.drop('const')

# Create a combined visualization: coefficients with significance indicators
plt.figure(figsize=(16, 14))
# Create the heatmap with custom colormap
heatmap_combined = sns.heatmap(coef_df, annot=True, cmap=custom_cmap,
                               fmt='.3f', linewidths=.5, annot_kws={"size": 16},
                               cbar_kws={'label': 'Coefficient Value'})

# Add asterisks to significant values
for i in range(len(coef_df.index)):
    for j in range(len(coef_df.columns)):
        if pval_df.iloc[i, j] < 0.001:
            plt.text(j + 0.5, i + 0.85, '***', ha='center', color='black', fontweight='bold', fontsi
        elif pval_df.iloc[i, j] < 0.01:
            plt.text(j + 0.5, i + 0.85, '**', ha='center', color='black', fontweight='bold', fontsiz
        elif pval_df.iloc[i, j] < 0.05:
            plt.text(j + 0.5, i + 0.85, '*', ha='center', color='black', fontweight='bold', fontsize

plt.title('Coefficient Heatmap with Significance Indicators', fontsize=24)
plt.xlabel('Project Success Outcomes', fontsize=22)
plt.ylabel('Predictors', fontsize=22)
plt.xticks(fontsize=18)
plt.yticks(fontsize=18)
plt.figtext(0.01, 0.01, '* p<0.05, ** p<0.01, *** p<0.001', ha='left', fontsize=16)

# Adjust colorbar font size
cbar = heatmap_combined.collections[0].colorbar
cbar.set_label('Coefficient Value', rotation=270, labelpad=30, fontsize=20)
cbar.ax.tick_params(labelsize=16)

plt.tight_layout()
plt.show()
```

```python
# P-Value Heatmap / Significance

# Set global font size
plt.rcParams.update({'font.size': 14})

# Define X and y
X = df[['x1_Empathy', 'x2_Automation', 'x3_Vision', 'x4_SustEthics', 'x5_ReqSource',
        'x6_Growth', 'x7_Human', 'x8_Scope', 'x9_Governance', 'x10_Strategy',
        'x11_IssueRes', 'x12_Retrospective', 'x13_Impediments', 'x14_Integration']]
Y = df[['y1_Schedule', 'y2_Cost', 'y3_Quality', 'y4_Scope', 'y5_Customer']]

# Add constant to X for intercept
X = sm.add_constant(X)

# Create a DataFrame to store p-values
pval_df = pd.DataFrame(index=X.columns)

# Fit models and extract p-values
for dep_var in Y.columns:
    model = sm.OLS(df[dep_var], X).fit()
    pval_df[dep_var] = model.pvalues

# Remove the constant row
pval_df = pval_df.drop('const')
colors = ['#006400', '#32CD32', '#FFFFE0']  # Dark Green, Light Green, Light Yellow
custom_cmap = LinearSegmentedColormap.from_list('GreenYellowSignificance', colors)

# Create a mask for non-significant values (to highlight significant ones)
mask = pval_df >= 0.05

# Set up the figure with a light background
plt.figure(figsize=(16, 14), facecolor='#FFFFF0')  # Light cream background
ax = plt.axes()
ax.set_facecolor('#FFFFF0')

# Create the p-value heatmap
heatmap = sns.heatmap(pval_df, annot=True, fmt='.4f',
                      cmap=custom_cmap, mask=mask,
                      linewidths=1.5, vmin=0, vmax=0.1,
                      annot_kws={"size": 16, "weight": "bold"},
                      cbar_kws={'label': 'p-value',
                                'ticks': [0, 0.01, 0.05, 0.1]})

# Add a second heatmap layer for non-significant values with a yellow palette
yellow_cmap = LinearSegmentedColormap.from_list('YellowShades', ['#FFFACD', '#FFD700'])  # Light ye
sns.heatmap(pval_df, annot=True, fmt='.4f',
            cmap=yellow_cmap, mask=~mask,
            linewidths=1.5, vmin=0, vmax=1.0,
            annot_kws={"size": 16},
            cbar=False)

# Add significance indicators with star symbols
for i in range(len(pval_df.index)):
    for j in range(len(pval_df.columns)):
        if pval_df.iloc[i, j] < 0.001:
            plt.text(j + 0.5, i + 0.85, '***', ha='center', color='white', fontweight='bold', font
```

```
        elif pval_df.iloc[i, j] < 0.01:
            plt.text(j + 0.5, i + 0.85, '**', ha='center', color='white', fontweight='bold', fonts
        elif pval_df.iloc[i, j] < 0.05:
            plt.text(j + 0.5, i + 0.85, '*', ha='center', color='white', fontweight='bold', fontsiz

# Add a border to the heatmap
for _, spine in ax.spines.items():
    spine.set_visible(True)
    spine.set_linewidth(2)
    spine.set_color('#006400')  # Dark green border

# Enhance title and labels
plt.title('P-value Heatmap: Statistical Significance', fontsize=24, pad=20, fontweight='bold', colo
plt.xlabel('Project Success Outcomes', fontsize=22, labelpad=15, color='#006400')
plt.ylabel('Predictors', fontsize=22, labelpad=15, color='#006400')

# Increase tick label size
plt.xticks(fontsize=18, rotation=45)
plt.yticks(fontsize=18)

# Add legend with larger font
plt.figtext(0.01, 0.01, '* p<0.05, ** p<0.01, *** p<0.001', ha='left', fontsize=18, color='#0064
# Adjust colorbar font size
cbar = heatmap.collections[0].colorbar
cbar.set_label('p-value', rotation=270, labelpad=30, fontsize=20, color='#006400')
cbar.ax.tick_params(labelsize=16)

plt.tight_layout()
plt.show()
```

VIF Score:

```
# Dependent and independent variables
y1 = df["Y_Project_Success"]
X1 = df[["x1_Empathy", "x2_Automation", "x3_Vision", "x4_SustEthics",
        "x5_ReqSource", "x6_Growth", "x7_Human", "x8_Scope",
        "x9_Governance", "x10_Strategy", "x11_IssueRes", "x12_Retrospective",
        "x13_Impediments", "x14_Integration"]]

# Add constant
X1 = sm.add_constant(X1)
```

```
# Import the required function
from statsmodels.stats.outliers_influence import variance_inflation_factor

# Calculate VIF for each variable
vif_data = pd.DataFrame()
vif_data["Feature"] = X1.columns
vif_data["VIF"] = [variance_inflation_factor(X1.values, i) for i in range(X1.shape[1])]
print(vif_data)
```

Cook's Distance:

```
import statsmodels.api as sm
# Import the OLSInfluence class from statsmodels
from statsmodels.stats.outliers_influence import OLSInfluence

model = sm.OLS(y1, X1).fit()
influence = OLSInfluence(model)

# Cook's distance
cooks_d = influence.cooks_distance[0]
influential_points = np.where(cooks_d > 4 / len(X1))[0]  # rule of thumb
print("Influential observations:", influential_points)
```

Robust Linear Regression Model:

```
rlm_model = sm.RLM(y1, X1, M=sm.robust.norms.HuberT())
rlm_results = rlm_model.fit()
print(rlm_results.summary())
```

Bootstrap mean & Influential observations:

```
from sklearn.utils import resample

coefs = []
for i in range(1000):  # 1000 bootstrap samples
    sample = resample(df)
    y_sample = sample["Y_Project_Success"]
    X_sample = sm.add_constant(sample[X1.columns.drop("const")])
    model_bs = sm.OLS(y_sample, X_sample).fit()
    coefs.append(model_bs.params["x4_SustEthics"])

# Confidence interval for Sustainability coefficient
print("Bootstrap mean:", np.mean(coefs))
print("95% CI:", np.percentile(coefs, [2.5, 97.5]))
```

```
influence = model.get_influence()
(c, p) = influence.cooks_distance

plt.stem(range(len(c)), c, markerfmt=",")
plt.title("Cook's Distance for Outlier Detection")
plt.xlabel("Observation index")
plt.ylabel("Cook's Distance")
plt.show()
```

```
import numpy as np
import pandas as pd
import statsmodels.api as sm

influence = model.get_influence()
cooks_d, pvals = influence.cooks_distance

# Convert to dataframe with observation indices
cooks_df = pd.DataFrame({
    "obs": np.arange(len(cooks_d)),
```

```
    "cooks_d": cooks_d
})

# Apply threshold
threshold = 4 / len(cooks_d)
influential_points = cooks_df[cooks_df["cooks_d"] > threshold]

print("Threshold:", threshold)
print("Influential observations:")
print(influential_points)
```

LASSO Regression Model:

```
# LASSO Regression
# (As VIF score shows multicolinearity, we will use Lasso Regresstion
# This will automatically selects important variables and shrinks/zero-out out others)

from sklearn.linear_model import LassoCV

# Check if 'const' column exists before dropping it
if 'const' in X.columns:
    X_lasso = X.drop('const', axis=1)
else:
    X_lasso = X  # Use X as is if 'const' doesn't exist

model_a = LassoCV(cv=5).fit(X_lasso, Y)
print(model_a.coef_)
```

```
from sklearn.linear_model import LassoCV

# Drop 'const' if present
if 'const' in X.columns:
    X_lasso = X.drop('const', axis=1)
else:
    X_lasso = X

# Fit LassoCV
model_a = LassoCV(cv=5).fit(X_lasso, Y)

# Print coefficients with feature names
for feature, coef in zip(X_lasso.columns, model_a.coef_):
    print(f"{feature} = {coef:.4f}")
```

```
# Lasso Coefficient Path:

from sklearn.linear_model import lasso_path
from sklearn.preprocessing import StandardScaler

# Define the predictors and response
X = df[['x1_Empathy', 'x2_Automation', 'x3_Vision', 'x4_SustEthics', 'x5_ReqSource',
        'x6_Growth', 'x7_Human', 'x8_Scope', 'x9_Governance', 'x10_Strategy',
        'x11_IssueRes', 'x12_Retrospective', 'x13_Impediments', 'x14_Integration']]
y = df['Y_Project_Success']
```

```
# Standardize predictors
scaler = StandardScaler()
X_std = scaler.fit_transform(X)

# Compute Lasso path
alphas_lasso, coefs_lasso, _ = lasso_path(X_std, y)

# Plot the path
plt.figure(figsize=(10, 6))
colors = plt.cm.viridis(np.linspace(0, 1, coefs_lasso.shape[0]))

for i in range(coefs_lasso.shape[0]):
    plt.plot(np.log10(alphas_lasso), coefs_lasso[i], label=X.columns[i], linewidth=2)

plt.xlabel('log10(alpha)', fontsize=12)
plt.ylabel('Coefficients', fontsize=12)
plt.title('Lasso Coefficient Path', fontsize=14)
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')
plt.grid(True)
plt.tight_layout()
plt.show()
```

One-Way ANOVA:

```
# Hypothesis Testing each independent variable (xi)
# One - Way ANOVA
# for x1_Empathy
# Test if different levels of x1_Empathy (different categoriesf x1 on Linkert Scale)
# impact project success
anova_model = smf.ols('Y_Project_Success ~ C(x1_Empathy)', data=df).fit()
anova_results = sm.stats.anova_lm(anova_model, typ=2)
print(anova_results)
```

```
# Test if different levels of the x2_Automation impact project success:

anova_model = smf.ols('Y_Project_Success ~ C(x2_Automation)', data=df).fit()
anova_results = sm.stats.anova_lm(anova_model, typ=2)
print(anova_results)
```

```
# Test if different levels of the x3_Vision impact project success:

anova_model = smf.ols('Y_Project_Success ~ C(x3_Vision)', data=df).fit()
anova_results = sm.stats.anova_lm(anova_model, typ=2)
print(anova_results)
```

```
# Test if different levels of x4_SustEthics impact project success:

anova_model = smf.ols('Y_Project_Success ~ C(x4_SustEthics)', data=df).fit()
anova_results = sm.stats.anova_lm(anova_model, typ=2)
print(anova_results)
```

```
# Test if different levels of x5_ReqSource impact project success:

anova_model = smf.ols('Y_Project_Success ~ C(x5_ReqSource)', data=df).fit()
anova_results = sm.stats.anova_lm(anova_model, typ=2)
print(anova_results)
```

```
# Test if different levels of x6_Growth impact project success:

anova_model = smf.ols('Y_Project_Success ~ C(x6_Growth)', data=df).fit()
anova_results = sm.stats.anova_lm(anova_model, typ=2)
print(anova_results)
```

```
# Test if different levels of x7_Human impact project success:

anova_model = smf.ols('Y_Project_Success ~ C(x7_Human)', data=df).fit()
anova_results = sm.stats.anova_lm(anova_model, typ=2)
print(anova_results)
```

```
# Test if different levels of x8_Scope impact project success:

anova_model = smf.ols('Y_Project_Success ~ C(x8_Scope)', data=df).fit()
anova_results = sm.stats.anova_lm(anova_model, typ=2)
print(anova_results)
```

```
# Test if different levels of x9_Governance impact project success:

anova_model = smf.ols('Y_Project_Success ~ C(x9_Governance)', data=df).fit()
anova_results = sm.stats.anova_lm(anova_model, typ=2)
print(anova_results)
```

```
# Test if different levels of x10_Strategy impact project success:

anova_model = smf.ols('Y_Project_Success ~ C(x10_Strategy)', data=df).fit()
anova_results = sm.stats.anova_lm(anova_model, typ=2)
print(anova_results)
```

```
# Test if different levels of x11_IssueRes impact project success:

anova_model = smf.ols('Y_Project_Success ~ C(x11_IssueRes)', data=df).fit()
anova_results = sm.stats.anova_lm(anova_model, typ=2)
print(anova_results)
```

```
# Test if different levels of x12_Retrospective impact project success:

anova_model = smf.ols('Y_Project_Success ~ C(x12_Retrospective)', data=df).fit()
anova_results = sm.stats.anova_lm(anova_model, typ=2)
print(anova_results)
```

```
# Test if different levels of x13_Impediments impact project success:

anova_model = smf.ols('Y_Project_Success ~ C(x13_Impediments)', data=df).fit()
anova_results = sm.stats.anova_lm(anova_model, typ=2)
print(anova_results)
```

```
# Test if different levels of x14_Integration impact project success:

anova_model = smf.ols('Y_Project_Success ~ C(x14_Integration)', data=df).fit()
anova_results = sm.stats.anova_lm(anova_model, typ=2)
print(anova_results)
```

```python
# Test if different types of PM methods in "Method" column impact project success:
# Method 1 - One-Way ANOVA (if Method has 3 or more groups)

import statsmodels.api as sm
import statsmodels.formula.api as smf

anova_model = smf.ols('Y_Project_Success ~ C(Method)', data=df).fit()
anova_table = sm.stats.anova_lm(anova_model, typ=2)
print(anova_table)
```

```python
# Test if different types of PM methods in "Method" column impact project success:
# Method 2 - Multiple Linear Regression (for adjusting other variables)
# Here we want to test the effect of "Method" on Project success while controlling for
# other independent variables such as x1 to x14

model_c = smf.ols('Y_Project_Success ~ C(Method) + x1_Empathy   + x2_Automation + x3_Vision + x4_Su
print(model_c.summary())
```

Structural Equation Modelling (SEM):

```python
# Install required packages and libraries for SEM analysis:

# Install the semopy package
!pip install semopy
# Import model from package
from semopy import Model
```

```python
# SEM Analysis:

# SEM model definition:
desc = '''
project_success =~ y1_Schedule + y2_Cost + y3_Quality + y4_Scope + y5_Customer
project_success ~ x1_Empathy + x2_Automation + x3_Vision + x4_SustEthics + x5_ReqSource + x6_Growth
'''

# Build and fit the SEM model
sem_model = Model(desc)
sem_model.fit(df)

# Show SEM results
sem_results = sem_model.inspect()
print(sem_results)
```

Model Training: RF, XGBoost, MLP, SVR:

```python
# Prepare Data:

X = df[['x1_Empathy', 'x2_Automation', 'x3_Vision', 'x4_SustEthics', 'x5_ReqSource', 'x6_Growth', '
y = df['Y_Project_Success']  #The composite score from PCA
```

```python
# Install Xgboost package
!pip install xgboost
```

```python
# Train/ Test Split the dataset:

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
```

```python
# Import Standardize Features such as SVR, MLP

# Imports the StandardScaler class from scikit-learn. This is used to scale numerical data.
from sklearn.preprocessing import StandardScaler

# Creates a scaler object.
scaler = StandardScaler()

# Learn scaling parameters from training data and apply them
X_train_scaled = scaler.fit_transform(X_train)

# Apply same scaling to test data (no re-learning)
X_test_scaled = scaler.transform(X_test)
```

Random Forest Regressor Model - Train & Test Model:

```python
# 1. Random Forest Model

# Import Libraries
from sklearn.metrics import mean_squared_error, r2_score
import numpy as np
from sklearn.ensemble import RandomForestRegressor

# Train the RF model
rf = RandomForestRegressor(n_estimators=100, random_state=42)
rf.fit(X_train, y_train)

# Make predictions on both training and test data
y_train_pred_rf = rf.predict(X_train)
y_test_pred_rf = rf.predict(X_test)

# Evaluation function
def evaluate(y_true, y_pred, name):
    print(f"{name} Results:")
    print(f"  R² Score: {r2_score(y_true, y_pred):.4f}")
    # Calculate RMSE manually by taking the square root of MSE
    print(f"  RMSE: {np.sqrt(mean_squared_error(y_true, y_pred)):.4f}")
    print("-" * 30)

# Evaluate RF on training data
evaluate(y_train, y_train_pred_rf, "Random Forest (Training)")

# Evaluate RF on test data
evaluate(y_test, y_test_pred_rf, "Random Forest (Test)")
```

```python
# RF Model - Actual vs Predicted Plot to shows how close predictions are to true values:

import matplotlib.pyplot as plt

plt.scatter(y_test, y_test_pred_rf, color='red', alpha=0.6)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')
plt.xlabel('Actual Project Success')
plt.ylabel('Predicted Project Success')
plt.title('Random Forest Model: Actual vs Predicted')
plt.show()
```

```python
# RF Model - Residual plot to verify model stability and identify any bias:

residuals = y_test - y_test_pred_rf
plt.scatter(y_test_pred_rf, residuals, color='blue', alpha=0.6)
plt.axhline(0, color='red', linestyle='--')
plt.xlabel('Predicted Values')
plt.ylabel('Residuals')
plt.title('Residual Plot for Random Forest Model')
plt.show()
```

XGBoost Regressor Model - Train & Test Model:

```python
# 2. XGBoost Regressor

# Import Libraries
from xgboost import XGBRegressor
from sklearn.metrics import mean_squared_error, r2_score
import numpy as np

# Train the XGBoost model
xgb = XGBRegressor(n_estimators=100, random_state=42)
xgb.fit(X_train, y_train)

# Make predictions on both training and test data
y_train_pred_xgb = xgb.predict(X_train)
y_test_pred_xgb = xgb.predict(X_test)

# Evaluation function
def evaluate(y_true, y_pred, name):
    print(f"{name} Results:")
    print(f"  R² Score: {r2_score(y_true, y_pred):.4f}")
    # Calculate RMSE manually by taking the square root of MSE
    print(f"  RMSE: {np.sqrt(mean_squared_error(y_true, y_pred)):.4f}")
    print("-" * 30)

# Evaluate RF on training data
evaluate(y_train, y_train_pred_xgb, "XGBoost (Training)")

# Evaluate RF on test data
evaluate(y_test, y_test_pred_xgb, "XGBoost (Test)")
```

```python
# XGBoost Model - Actual vs Predicted Plot to shows how close predictions are to true values:

import matplotlib.pyplot as plt

plt.scatter(y_test, y_test_pred_xgb, color='green', alpha=0.6)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')
plt.xlabel('Actual Project Success')
plt.ylabel('Predicted Project Success')
plt.title('XGBoot Model: Actual vs Predicted')
plt.show()
```

```python
# XGBoost Model - Residual plot to verify model stability and identify any bias:

residuals = y_test - y_test_pred_xgb
plt.scatter(y_test_pred_xgb, residuals, color='red', alpha=0.6)
plt.axhline(0, color='red', linestyle='--')
plt.xlabel('Predicted Values')
plt.ylabel('Residuals')
plt.title('Residual Plot for XGBoost Model')
plt.show()
```

Support Vector Regressor (SVR) Model:

```python
# 3. Support Vector Regressor (SVR)
from sklearn.svm import SVR
from sklearn.metrics import mean_squared_error, r2_score
import numpy as np

# Train the SVR model
svr = SVR(kernel='rbf')
svr.fit(X_train_scaled, y_train)

# Make predictions on both training and test data
y_train_pred_svr = svr.predict(X_train_scaled)
y_test_pred_svr = svr.predict(X_test_scaled)

# Evaluation function
def evaluate(y_true, y_pred, name):
    print(f"{name} Results:")
    print(f"  R² Score: {r2_score(y_true, y_pred):.4f}")
    # Calculate RMSE manually by taking the square root of MSE
    print(f"  RMSE: {np.sqrt(mean_squared_error(y_true, y_pred)):.4f}")
    print("-" * 30)

# Evaluate SVR on training data
evaluate(y_train, y_train_pred_svr, "SVR (Training)")

# Evaluate SVR on test data
evaluate(y_test, y_test_pred_svr, "SVR (Test)")
```

```python
# Actual vs Predicted Plot to shows how close predictions are to true values:

import matplotlib.pyplot as plt

plt.scatter(y_test, y_test_pred_svr, color='blue', alpha=0.6)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')
plt.xlabel('Actual Project Success')
plt.ylabel('Predicted Project Success')
plt.title('SVR Model: Actual vs Predicted')
plt.show()
```

```python
# residual plot to verify model stability and identify any bias:

residuals = y_test - y_test_pred_svr
plt.scatter(y_test_pred_svr, residuals, color='purple', alpha=0.6)
plt.axhline(0, color='red', linestyle='--')
plt.xlabel('Predicted Values')
plt.ylabel('Residuals')
plt.title('Residual Plot for SVR Model')
plt.show()
```

Multi-layer Perceptron (MLP) Regressor Model:

```python
# 4. Multi-layer Perceptron (MLP) Regressor

from sklearn.neural_network import MLPRegressor
from sklearn.metrics import mean_squared_error, r2_score
import numpy as np

# Train the MLP model
mlp = MLPRegressor(hidden_layer_sizes=(64, 32), max_iter=1000, random_state=42)
mlp.fit(X_train_scaled, y_train)

# Make predictions on both training and test data
y_train_pred_mlp = mlp.predict(X_train_scaled)
y_test_pred_mlp = mlp.predict(X_test_scaled)


# Evaluation function
def evaluate(y_true, y_pred, name):
    print(f"{name} Results:")
    print(f"  R² Score: {r2_score(y_true, y_pred):.4f}")
    # Calculate RMSE manually by taking the square root of MSE
    print(f"  RMSE: {np.sqrt(mean_squared_error(y_true, y_pred)):.4f}")
    print("-" * 30)

# Evaluate SVR on training data
evaluate(y_train, y_train_pred_mlp, "MLP (Training)")

# Evaluate SVR on test data
evaluate(y_test, y_test_pred_mlp, "MLP (Test)")
```

```python
# MLP Regressor: - Actual vs Predicted Plot to shows how close predictions are to true values:

import matplotlib.pyplot as plt
```

```
plt.scatter(y_test, y_test_pred_mlp, color='green', alpha=0.6)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')
plt.xlabel('Actual Project Success')
plt.ylabel('Predicted Project Success')
plt.title('MLP Model: Actual vs Predicted')
plt.show()
```

```
# Residual plot to verify model stability and identify any bias: MLP

residuals = y_test - y_test_pred_mlp
plt.scatter(y_test_pred_mlp, residuals, color='brown', alpha=0.6)
plt.axhline(0, color='red', linestyle='--')
plt.xlabel('Predicted Values')
plt.ylabel('Residuals')
plt.title('Residual Plot for MLP Model')
plt.show()
```

Random Forest Model – Feature Importance Discovery:

```
# Random Forest Model:
# Feature Importance to discover which variables matter most

import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np

# Sort feature importances
feat_importances = pd.Series(rf.feature_importances_, index=X.columns).sort_values(ascending=True)

# Plot with Seaborn
plt.figure(figsize=(10, 6))
sns.barplot(x=feat_importances, y=feat_importances.index, palette="viridis")

# Add labels and values
plt.title("Feature Importance (Random Forest)", fontsize=14, fontweight='bold')
plt.xlabel("Importance Score")
plt.ylabel("Features")

# Add text labels on bars
for i, v in enumerate(feat_importances):
    plt.text(v + 0.001, i, f"{v:.3f}", color='black', va='center')

plt.tight_layout()
plt.show()
```

XGBoost Model – Feature Importance Discovery:

```
# XGBoost Model:
# Feature Importance to discover which variables matter most

import pandas as pd
import xgboost as xgb
```

```
# Create and fit an XGBoost model
model = xgb.XGBRegressor()  # or XGBClassifier() depending on task
model.fit(X, y)  # Assuming X and y are features and target variables

# Get feature importances from the fitted model (not the module)
importances = pd.Series(model.feature_importances_, index=X.columns)

# Sort and print top features
importances = importances.sort_values(ascending=False)
print(importances)
```

```
# Create figure with appropriate size
plt.figure(figsize=(12, 8))

# Get top 10 features
top_features = importances.head(10).sort_values(ascending=True)

# Create horizontal bar plot with a different color palette
bars = sns.barplot(x=top_features.values, y=top_features.index, palette="coolwarm")

# Add value labels to the bars
for i, v in enumerate(top_features.values):
    plt.text(v + v*0.01, i, f"{v:.4f}", va='center', fontweight='bold')

# Customize the plot
plt.title("Top 10 Feature Importances (XGBoost)", fontsize=16, fontweight='bold', pad=20)
plt.xlabel("Importance Score", fontsize=14, labelpad=10)
plt.ylabel("Features", fontsize=14, labelpad=10)

# Add a subtle grid for better readability
plt.grid(axis='x', linestyle='--', alpha=0.6)

# Add a light border around the plot
for spine in plt.gca().spines.values():
    spine.set_edgecolor('#CCCCCC')

# Adjust layout
plt.tight_layout()

# Show the plot
plt.show()
```

SHAP (SHapley Additive exPlanations) Model:

```
# Install SHAP Package

!pip install shap
```

```
# Use SHAP to Interpreat MLP Regressor - Train MLP Regressor:

from sklearn.neural_network import MLPRegressor
from sklearn.model_selection import train_test_split

# Split dataset
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```python
# Train MLP
mlp = MLPRegressor(hidden_layer_sizes=(100, 50), max_iter=1000, random_state=42)
mlp.fit(X_train, y_train)
```

```
MLPRegressor
MLPRegressor(hidden_layer_sizes=(100, 50), max_iter=1000, random_state=42)
```

```python
# Use SHAP to explain prediction:

import shap
import numpy as np

# Convert to numpy array
X_train_np = X_train.values if isinstance(X_train, pd.DataFrame) else X_train
X_test_np = X_test.values if isinstance(X_test, pd.DataFrame) else X_test

# Use SHAP Kernel Explainer, since MLP is not tree-based
# Sample to reduce compute time
explainer = shap.KernelExplainer(mlp.predict, shap.sample(X_train_np, 100))
# Explain a few predictions
shap_values = explainer.shap_values(X_test_np[:50])  # Explain first 50 rows

# Plot summary plot
shap.summary_plot(shap_values, X_test_np[:50], feature_names=X.columns)
```

```python
# Check the number of samples in X_test
print(f"Number of samples in X_test: {len(X_test)}")

# Pick a valid sample index (for example: the first test case)
sample_index = 0  # Using index 0 instead of 10

# Compute SHAP values again (if not already done)
explainer = shap.Explainer(model, X_train)
shap_values = explainer(X_test)

# Create the force plot with increased width
shap.initjs()  # enable JS-based visualization in Jupyter

# Increase the plot width to show all variables
shap.force_plot(
    explainer.expected_value,        # base value (average prediction)
    shap_values[sample_index].values, # SHAP values for this sample
    X_test.iloc[sample_index],       # input features for this sample
    feature_names=X.columns,
    figsize=(20, 3),                 # Increase width (20) while keeping height (3) reasonable
    matplotlib=True                  # Use matplotlib renderer for better control
)
```

```python
shap.dependence_plot(
    "x1_Empathy",
    shap_values.values,
    X_test_scaled,
    feature_names=X.columns,
    interaction_index="x2_Automation"
)
```

```python
shap.dependence_plot(
    "x12_Retrospective",
    shap_values.values,
    X_test_scaled,
    feature_names=X.columns,
    interaction_index="x6_Growth"
)
```

```python
shap.dependence_plot(
    "x3_Vision",
    shap_values.values,
    X_test_scaled,
    feature_names=X.columns,
    interaction_index="x5_ReqSource"
)
```

```python
shap.dependence_plot(
    "x4_SustEthics",
    shap_values.values,
    X_test_scaled,
    feature_names=X.columns,
    interaction_index="x13_Impediments"
)
```

```python
# Calculate R-squared and RMSE for both training and test sets
from sklearn.metrics import r2_score, mean_squared_error
import numpy as np

# Function to calculate and format metrics
def calculate_metrics(y_true, y_pred, dataset_name=""):
    r2 = r2_score(y_true, y_pred)
    rmse = np.sqrt(mean_squared_error(y_true, y_pred))

    print(f"SHAP ({dataset_name}) Results:")
    print(f"  R² Score: {r2:.4f}")
    print(f"  RMSE: {rmse:.4f}")
    print("-" * 30)

    return r2, rmse

# Get predictions for both training and test sets
y_train_pred = mlp.predict(X_train)
y_test_pred = mlp.predict(X_test)

# Calculate and display metrics for training set
train_r2, train_rmse = calculate_metrics(y_train, y_train_pred, "Training")

# Calculate and display metrics for test set
test_r2, test_rmse = calculate_metrics(y_test, y_test_pred, "Test")

# Save these metrics to a dictionary or DataFrame
metrics = {
    'Training': {'R2': train_r2, 'RMSE': train_rmse},
    'Test': {'R2': test_r2, 'RMSE': test_rmse}
}
```

```python
# Calculate R-squared and RMSE for model evaluation
from sklearn.metrics import r2_score, mean_squared_error
import numpy as np

# Visualize actual vs predicted values
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_pred, alpha=0.5)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
plt.title('Actual vs Predicted Values')
plt.grid(True)
plt.show()

# Plot residuals
residuals = y_test - y_pred
plt.figure(figsize=(10, 6))
plt.scatter(y_pred, residuals, alpha=0.5)
plt.hlines(y=0, xmin=y_pred.min(), xmax=y_pred.max(), colors='r', linestyles='--')
plt.xlabel('Predicted Values')
plt.ylabel('Residuals')
plt.title('Residual Plot')
plt.grid(True)
plt.show()
```

```python
# Bar plot of global feature importance using SHAP
shap.summary_plot(shap_values, X_test_np[:50], feature_names=X.columns, plot_type='bar')

# Explain a single instance
shap.initjs()
shap.force_plot(explainer.expected_value, shap_values[0], features=X_test_np[0], feature_names=X.co
```

LIME (Local Interpretable Model-agnostic Explanations) Model:

```python
# Install LIME package

!pip install lime
```

```python
# Use LIME to Interpreat MLP Regressor - Train MLP Regressor:

from sklearn.neural_network import MLPRegressor
from sklearn.model_selection import train_test_split
import pandas as pd

# Split data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train MLP
mlp = MLPRegressor(hidden_layer_sizes=(100, 50), max_iter=1000, random_state=42)
mlp.fit(X_train, y_train)
```

```
MLPRegressor
MLPRegressor(hidden_layer_sizes=(100, 50), max_iter=1000, random_state=42)
```

```
# Lime Feature Importance:

import lime
import lime.lime_tabular
import numpy as np

# Create explainer
explainer = lime.lime_tabular.LimeTabularExplainer(
    training_data=np.array(X_train),
    feature_names=X.columns.tolist(),
    mode='regression'
)

# Pick a sample to explain (for example 1st test sample)
i = 0
exp = explainer.explain_instance(
    data_row=X_test.iloc[i],
    predict_fn=mlp.predict,
    num_features=len(X.columns)
)

# Show explanation in notebook
exp.show_in_notebook(show_table=True)

# Or print to console
print(exp.as_list())
```

```
# Calculate R-squared and RMSE for LIME model evaluation
from sklearn.metrics import r2_score, mean_squared_error
import numpy as np

# Function to calculate and format metrics
def calculate_metrics(y_true, y_pred, dataset_name=""):
    r2 = r2_score(y_true, y_pred)
    rmse = np.sqrt(mean_squared_error(y_true, y_pred))

    print(f"LIME ({dataset_name}) Results:")
    print(f"  R² Score: {r2:.4f}")
    print(f"  RMSE: {rmse:.4f}")
    print("-" * 30)

    return r2, rmse

# Get predictions for both training and test sets
y_train_pred = mlp.predict(X_train)
y_test_pred = mlp.predict(X_test)

# Calculate and display metrics for training set
train_r2, train_rmse = calculate_metrics(y_train, y_train_pred, "Training")
```

```python
# Calculate and display metrics for test set
test_r2, test_rmse = calculate_metrics(y_test, y_test_pred, "Test")

# Optional: Visualize actual vs predicted values
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_test_pred, alpha=0.5)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
plt.title('LIME Model: Actual vs Predicted Values')
plt.grid(True)
plt.show()

# Optional: Store metrics in a dictionary for later comparison
lime_metrics = {
    'Training': {'R2': train_r2, 'RMSE': train_rmse},
    'Test': {'R2': test_r2, 'RMSE': test_rmse}
}
```

Supervised Learning ML Model - Decision Tree:

```python
from sklearn.tree import DecisionTreeRegressor, plot_tree
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt

# Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train Decision Tree
tree = DecisionTreeRegressor(max_depth=4, random_state=42)
tree.fit(X_train, y_train)

# Predict
y_pred = tree.predict(X_test)

# Evaluate
print("R2 Score:", r2_score(y_test, y_pred))
print("MSE:", mean_squared_error(y_test, y_pred))

# Plot the tree with increased font sizes
plt.figure(figsize=(20,10))
plot_tree(
    tree,
    feature_names=X.columns,
    filled=True,
    rounded=True,
    fontsize=14,  # Increased font size for tree text
    precision=2   # Limit decimal places for cleaner display
)
plt.title("Decision Tree for Predicting Project Success", fontsize=18)
plt.tight_layout()  # Ensures everything fits well
plt.show()
```

Unsupervised Learning ML Model - Rule Mining:

```
# Install the mlxtend package

!pip install mlxtend
```

```
from mlxtend.frequent_patterns import apriori, association_rules
from mlxtend.preprocessing import TransactionEncoder
import pandas as pd

# Convert all features to categorical/ binned (2 bins)
X_binned = X.copy()
for col in X_binned.columns:
    X_binned[col] = pd.qcut(X_binned[col], q=2, labels=["Low", "High"], duplicates='drop')

# Convert dataframe to list of transactions
transactions = X_binned.apply(lambda row: [f"{col}_{val}" for col, val in row.items()], axis=1)

# Convert to binary matrix
te = TransactionEncoder()
te_ary = te.fit(transactions).transform(transactions)
df_te = pd.DataFrame(te_ary, columns=te.columns_)

# Apply Apriori with higher min_support and limit max_len to 2 for speed
frequent_itemsets = apriori(df_te, min_support=0.15, use_colnames=True, max_len=2)

print(f"Number of frequent itemsets found: {len(frequent_itemsets)}")

# Generate rules with lower confidence threshold
rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.5)
rules = rules.sort_values(by='lift', ascending=False)

# View top rules
if len(rules) > 0:
    print(rules[['antecedents', 'consequents', 'support', 'confidence', 'lift']].head(10))
else:
    print("No association rules found. Try adjusting min_support or min_threshold further.")
```