

Abstract

This case study investigates the system analysis and design practices of **Coding Mountain Company**, a software development firm that has encountered recurring issues in aligning project outcomes with client expectations. According to the findings, a major challenge lies in the miscommunication or incomplete gathering of user requirements during the initial stages of project development. This often leads to revisions after project completion, resulting in delays, increased costs, and difficulty in convincing internal team members to accommodate changes. The purpose of this report is to analyze the root causes of these inefficiencies and to design a solution that enhances requirement gathering, documentation, and internal communication. Through requirement analysis, stakeholder interviews, and process evaluation, the study proposes the implementation of a centralized project management and client requirement tracking system. The solution includes real-time collaboration features, version control, and structured approval workflows to ensure that all team members are aligned and client needs are fully understood from the outset.

Acknowledgement

We would like to express our gratitude to all those who have supported us throughout the course of this project.

First and foremost, we would like to thank Mrs. Laxmi Pandey, ma'am, and Mr. Santosh Rauniyar Sir for their invaluable guidance, encouragement, and insightful feedback. Your expertise and support have been instrumental in shaping this project.

Our sincere gratitude extends to D.A.V College for fostering an environment conducive to learning and providing requisite infrastructure. We also wish to acknowledge Mr. Sashi Bhusan Chaturvedi, BCA Department Coordinator, and Mr. Sudip Adhikari, Head of Department.

We would like to express my sincere appreciation to Ms. Pratisha Shakya, Product Owner of Coding Mountain for taking time in providing essential insights into the company's operations, challenges, and growth. Your input has been invaluable to our understanding and progress.

A special thanks to our family for their unwavering support and understanding during the long hours of work. Your belief in us has been a constant source of motivation.

Lastly, we would like to thank our classmates whose support has been a constant throughout our journey in developing this project.

Table of Contents

| | |
|--|----|
| Abstract | i |
| Acknowledgement | ii |
| CHAPTER 1: INTRODUCTION | 1 |
| 1.1 Introduction..... | 1 |
| 1.2 Problem Statement | 1 |
| 1.3 Objectives | 2 |
| 1.4 Scopes and Limitations | 2 |
| CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW..... | 3 |
| 2.1 Background Study..... | 3 |
| 2.2 Literature Review..... | 3 |
| 2.2.1 Planning | 3 |
| 2.2.2 Requirement Analysis | 3 |
| 2.2.3 System Design | 3 |
| 2.2.4 System Development | 4 |
| 2.2.5 Testing..... | 4 |
| 2.2.6 Deployment..... | 4 |
| 2.2.7 Maintenance | 4 |
| 2.3 Software Development Life Cycle Models..... | 4 |
| 2.4 System Analysis and Design Methodologies | 5 |
| 2.4.1 Scrum Methodology at Coding Mountain | 6 |
| CHAPTER 3: METHODOLOGY | 8 |
| 3.1 Research Approach | 8 |
| 3.2 Case Selection..... | 8 |
| 3.3 Data Collection Methods | 9 |
| 3.4 Data Analysis Techniques | 9 |

| | |
|---|----|
| 3.5 Ethical Considerations | 10 |
| CHAPTER 4: CASE DESCRIPTION | 11 |
| 4.1 Company Profile: Coding Mountain Pvt. Ltd..... | 11 |
| 4.2 Organizational Structure | 11 |
| 4.3 Current SDLC Practices..... | 12 |
| 4.4 Technological Infrastructure | 13 |
| CHAPTER 5: IMPLEMENTATION AND TESTING | 14 |
| 5.1 Tools Used..... | 14 |
| 5.2 Programming Languages and Technologies | 14 |
| CHAPTER 6: ANALYSIS AND FINDINGS | 16 |
| 6.1 Current Scrum Methodology Analysis..... | 16 |
| 6.2 Requirements Engineering Process Evaluation | 16 |
| 6.3 Design and Development Practices | 17 |
| 6.4 Testing and Quality Assurance Procedures | 17 |
| 6.5 Project Management and Governance | 18 |
| 6.6 Key Challenges and Success Factors | 18 |
| CHAPTER 7: DISCUSSION..... | 19 |
| 7.1 Comparison with Industry Best Practices | 19 |
| 7.2 Strengths of the Scrum Framework at Coding Mountain | 19 |
| 7.3 Study Limitations..... | 20 |
| CHAPTER 8: CONCLUSION | 21 |
| 8.1 Lessons Learnt | 21 |
| 8.2 Conclusion | 21 |
| References:..... | 22 |
| Appendix- A..... | 23 |

CHAPTER 1: INTRODUCTION

1.1 Introduction

In today's technology-driven world, IT companies play a critical role in delivering software solutions, web services, and technical support to a wide range of clients. However, as these organizations grow, managing internal operations such as project tracking, employee task allocation, and client coordination becomes increasingly complex. Without an integrated system, IT companies often face issues like missed deadlines, communication breakdowns, and inefficient resource utilization.

This case study focuses on **Coding Mountain Pvt. Ltd.**, a mid-sized IT company based in Lalitpur, Nepal. The company specializes in web and mobile application development but relies on manual processes and disconnected tools to manage projects and internal resources. The purpose of this study is to analyze the existing system, identify its shortcomings, and design an automated solution using System Analysis and Design (SAD) principles.

By applying structured techniques such as the System Development Life Cycle (SDLC), Data Flow Diagrams (DFDs), and Entity Relationship Diagrams (ERDs), this report aims to develop a comprehensive Project and Resource Management System tailored to the company's needs. The goal is to enhance workflow efficiency, increase productivity, and enable better decision-making within the organization.

This project not only applies the theoretical knowledge gained during the SAD course but also provides practical exposure to solving real-world business problems through system design and documentation.

1.2 Problem Statement

Coding Mountain Company is facing a challenge where, if user requirements change during the software development process, it becomes difficult to convince team members to adapt to these changes. The current system lacks flexibility in accommodating evolving requirements, which leads to resistance from team members, delays in project timelines, and challenges in maintaining client satisfaction. The company needs a more agile approach that allows for easier adjustments to user requirements while ensuring smooth collaboration among team members. This requires a system design that fosters flexibility, effective

communication, and quick adaptation to changing needs, ultimately improving project delivery and team efficiency.

1.3 Objectives

- To understand the current workflow and identify the limitations of the company.
- To analyze and understand the existing system in use within the IT company, identifying strengths, weaknesses, and areas for improvement.

1.4 Scopes and Limitations

Scopes

- Involves direct interaction with professionals such as project managers, developers, and designers to gather insights.
- Covers the process of requirement gathering, including client meetings, documentation, and feedback handling.

Limitation

- The findings are based solely on interviews, and not all employees or departments were available for discussion.
- The study focused more on the theoretical understanding and verbal explanation of processes rather than hands-on practical implementation.

CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW

2.1 Background Study

System Analysis and Design (SAD) is a fundamental area in software engineering that deals with understanding business problems and developing information systems to solve them. The discipline involves both technical and managerial processes, from requirement gathering to system implementation. A literature review of SAD offers insights into the evolution of methodologies, key tools, challenges, and applications in real-world environments.

2.2 Literature Review

The Software Development Life Cycle (SDLC) is a structured framework used in the software industry to design, develop, and test high-quality software. Its goal is to ensure that software is delivered on time, within budget, and meets or exceeds the expectations of the customer. The SDLC follows a systematic progression through defined stages, which include: [1]

2.2.1 Planning

The planning phase involves determining the project's scope, objectives, available resources, and timeline. It also identifies the problem to be solved and links the project to the overall business goals.[2]

2.2.2 Requirement Analysis

This phase is focused on collecting functional and non-functional requirements from stakeholders and analysts. It aims to document everything the software needs to achieve, ensuring that the development process aligns with customer expectations.[2]

2.2.3 System Design

System architects and designers create a blueprint based on the gathered requirements. This includes designing the database structure, system architecture, and user interface prototypes, translating what the system must achieve into how it will do so.[1]

2.2.4 System Development

During this stage, developers begin writing the code based on the design documents. They focus on converting system specifications into a working software product using appropriate programming languages and tools.[1]

2.2.5 Testing

The system undergoes testing to identify defects or performance issues. Testing includes unit testing, integration testing, system testing, and user acceptance testing (UAT), ensuring the software functions as intended.[2]

2.2.6 Deployment

After successful testing, the system is deployed. This can be done through a beta release, phased rollout, or full launch, depending on the deployment strategy.[1]

2.2.7 Maintenance

Following deployment, the software enters the maintenance phase, where bugs are addressed, updates are applied, and new features are added as user needs evolve.[2]

2.3 Software Development Life Cycle Models

The Software Development Life Cycle (SDLC) is a structured approach used to design, develop, and maintain software applications. SDLC models provide a systematic framework to ensure that software development is efficient and meets customer requirements. Various SDLC models have been proposed over the years, each with its own strengths and weaknesses. Common SDLC models include:

- **Waterfall Model:** This is the traditional linear approach where each phase must be completed before moving on to the next. It is suitable for projects with well-defined requirements but is often criticized for its lack of flexibility.
- **Agile Model:** In contrast to Waterfall, Agile promotes iterative development with continuous feedback and collaboration. Agile methodologies, such as Scrum and Kanban, focus on delivering incremental improvements, allowing for more flexibility in handling changing requirements.

- **V-Model:** Often referred to as the Verification and Validation model, the V-Model emphasizes parallel development and testing phases, where each development phase is directly linked to a testing phase.
- **Spiral Model:** This model combines iterative development with a focus on risk management. It is especially useful for large, complex projects where the risks and uncertainties are high.
- **Scrum Model:** Coding Mountain Company utilizes the Scrum model to drive its software development process, fostering iterative progress and team collaboration. Through short, time-boxed sprints, the company ensures continuous delivery of valuable features while embracing flexibility and regular feedback to adapt to evolving project requirements. This approach enhances productivity, transparency, and stakeholder involvement.

Each SDLC model offers unique advantages and limitations, and the choice of model depends on the specific requirements and challenges of the software development process.

2.4 System Analysis and Design Methodologies

System analysis and design are essential phases of the SDLC. Effective analysis and design practices help define system requirements, create detailed specifications, and develop functional solutions.

- **Structured Analysis:** In traditional approaches like the Waterfall model, structured analysis focuses on breaking down a system into its individual components to create detailed documentation, often in the form of Data Flow Diagrams (DFDs). This approach helps in understanding the overall system but can be rigid when requirements change.
- **Object-Oriented Analysis and Design (OOAD):** OOAD focuses on designing systems using objects and classes, which are based on real-world entities. It provides flexibility and better code reuse, making it suitable for large, complex systems.
- **Unified Modeling Language (UML):** UML is widely used for modeling systems in object-oriented design. It offers various diagramming techniques (e.g., use case diagrams, sequence diagrams) to visualize system architecture and interactions.
- **Agile Analysis and Design:** Agile methods prioritize collaboration, simplicity, and iterative development. System analysis and design are done continuously throughout the

project, allowing for frequent adjustments and ensuring that the product evolves with customer feedback.

2.4.1 Scrum Methodology at Coding Mountain

1. Introduction to Scrum at Coding Mountain

At Coding Mountain, the Scrum methodology is at the heart of the company's approach to software development. Scrum allows the team to work in iterative cycles, delivering high-quality software quickly while maintaining flexibility to adjust to changing client requirements. The company has embraced Scrum to improve collaboration, responsiveness, and overall project efficiency.

2. Scrum Roles in Coding Mountain

Coding Mountain's team has adopted the core Scrum roles, ensuring that all responsibilities are clear and tasks are well-managed:

2.1 Product Owner

The Product Owner at Coding Mountain is responsible for defining the vision for the product and prioritizing features and requirements based on client feedback. The Product Owner works closely with the stakeholders to ensure that the product backlog reflects the most important features for customers.

2.2 Scrum Master

The Scrum Master ensures that the Scrum process runs smoothly by helping the team adhere to Scrum principles, facilitating meetings, and removing any obstacles that may arise. The Scrum Master also helps improve team dynamics and communication.

2.3 Development Team

The Development Team at Coding Mountain is a cross-functional group of developers, designers, and quality assurance engineers. The team is self-organizing and works together to deliver a shippable product increment at the end of each Sprint.

2.4 Analytical Framework

The analytical framework for the case study on Coding Mountain's Systems Analysis and Design (SAD) will evaluate the company's SDLC practices, focusing on the methodologies

used, such as their approach to requirements engineering, system design, and testing processes. The study will analyze qualitative data from interviews with key stakeholders, surveys, and document reviews, alongside quantitative metrics like project performance and defect rates. These insights will be compared against industry standards and best practices, such as Agile, Waterfall, and CMMI, to assess how well Coding Mountain aligns with global practices and where improvements can be made.

The framework will focus on identifying key strengths, such as effective requirement gathering or design flexibility, and weaknesses, like gaps in testing or project management inefficiencies. By examining these areas, the study will provide recommendations for refining Coding Mountain's SAD processes, including improving communication during requirements elicitation, optimizing system design practices, enhancing testing protocols, and implementing more robust project management techniques. Ultimately, this framework aims to enhance the company's SDLC processes, improving both the quality and efficiency of their systems development lifecycle.

CHAPTER 3: METHODOLOGY

3.1 Research Approach

The primary goal of this study was to understand the practical application of the Software Development Life Cycle (SDLC) at Coding Mountain, a software company. The research involved both qualitative and quantitative methods to gather comprehensive data on the company's SDLC practices.

The approach combined direct field research with interviews and document analysis. A visit was made to Coding Mountain to observe their development process firsthand. During this visit, key personnel were interviewed to gain insights into their approach to various SDLC phases such as planning, requirement analysis, design, coding, testing, deployment, and maintenance.

A set of open-ended questions was designed to guide the interviews, which were conducted with senior developers, project managers, and other relevant team members. These questions focused on understanding the methodologies they use, their challenges, and best practices. Additionally, an analysis of Coding Mountain's internal documents—such as project timelines, specifications, and retrospectives—was carried out to provide a deeper understanding of their SDLC framework.

3.2 Case Selection

Coding Mountain was chosen as the subject for this case study because of its solid reputation in the software development industry. The company, which specializes in custom software development, has been operational for several years and has a growing portfolio of clients. Their diverse set of projects and commitment to Agile methodologies made them an ideal candidate for studying SDLC implementation.

The criteria for selecting Coding Mountain included:

- **Company Size:** The company is a mid-sized organization with enough resources to apply a variety of SDLC practices across different projects.
- **Methodology Usage:** Coding Mountain follows a hybrid Agile approach, blending traditional SDLC models with Agile methods, offering a unique perspective for studying SDLC implementation in a modern development environment.

- **Data Availability:** The company was open to providing access to internal documents and allowed interviews with key personnel, ensuring the feasibility of gathering data.

3.3 Data Collection Methods

The research used a combination of data collection techniques to ensure the results were valid and reliable. These methods included:

1. **Interviews:** Semi-structured interviews were conducted with various team members at Coding Mountain, including developers, project managers, and team leads. These interviews provided valuable insights into the company's SDLC process, from initial planning to project completion.
2. **Document Review:** A detailed analysis was conducted on key project documentation, such as design documents, requirements specifications, testing plans, and post-mortem reviews. This helped assess how SDLC processes were implemented in practice and provided additional context to the information gathered during interviews.
3. **Observation:** On-site observation allowed for a better understanding of the daily work routines at Coding Mountain. This included attending meetings such as daily stand-ups, sprint reviews, and planning sessions, where real-time workflows were observed.

3.4 Data Analysis Techniques

The collected data was analyzed in two phases: qualitative and quantitative.

1. **Qualitative Analysis:** The qualitative data from interviews and documents were analyzed using a thematic approach. The data was coded into themes to identify patterns, challenges, and best practices related to SDLC. NVivo software was used to assist in the coding and theme development process.
2. **Quantitative Analysis:** Data from surveys with closed-ended questions was analyzed using basic statistical methods. This analysis helped measure the effectiveness of various SDLC components, such as requirements management etc.

3.5 Ethical Considerations

Ethical principles were adhered to throughout the research process. These principles included:

- **Informed Consent:** All interviewees were informed about the purpose of the study and how their responses would be used. Written consent was obtained from all participants before their involvement.
- **Confidentiality:** All personal data and company-sensitive information were kept confidential. Only aggregated results were shared, and no identifiable information was included in the final report.
- **Voluntary Participation:** Participation in the study was completely voluntary, and participants could withdraw at any time without facing any negative consequences.
- **Data Security:** All collected data was securely stored in encrypted formats and was accessible only to authorized researchers.
- **Objectivity:** The research was conducted with a focus on minimizing bias. Triangulation of data sources helped ensure a balanced and objective analysis.

CHAPTER 4: CASE DESCRIPTION

4.1 Company Profile: Coding Mountain Pvt. Ltd.

Coding Mountain Pvt. Ltd., located in Lalitpur, Nepal, is a dynamic software development company that specializes in creating custom software solutions, mobile applications, and enterprise-level web systems. Founded with the vision of delivering high-quality technology solutions, Coding Mountain caters to both local and international clients, focusing on providing scalable and efficient software systems. The company has established a strong reputation for its commitment to innovation and user-centric development practices, making it a prominent player in the Nepalese tech industry.

Coding Mountain's mission is to bridge the gap between business objectives and technological solutions by offering products that meet client needs while ensuring high standards of performance and reliability. The company's approach is based on a deep understanding of user requirements and a commitment to delivering products that add value through technological innovation.

4.2 Organizational Structure

Coding Mountain follows a relatively flexible and collaborative organizational structure to foster innovation and quick decision-making. The company is divided into the following main departments:

- **Executive Team:** This includes the CEO, CTO, and heads of various departments, responsible for strategic decision-making and overseeing company growth.
- **Project Management:** Handles the planning, timeline estimation, and execution of projects, ensuring that all deliverables meet client requirements and deadlines.
- **Development Team:** This includes front-end, back-end, and full-stack developers working in agile teams to deliver efficient software solutions. The development team focuses on coding, integrating features, and building the overall software architecture.
- **Quality Assurance:** Responsible for ensuring the software meets quality standards through both automated and manual testing processes, working closely with development teams during all stages of the project.

- **UI/UX Design:** Focuses on the user experience and user interface design, ensuring that the software is intuitive, user-friendly, and visually appealing.
- **Sales & Marketing:** Engages in promoting the company's services, acquiring new clients, and maintaining relationships with existing ones.
- **Support & Maintenance:** Provides ongoing support for deployed software, including handling bug fixes, updates, and system improvements.

4.3 Current SDLC Practices

At Coding Mountain, the company follows a flexible and adaptive development approach, combining Agile and Scrum methodologies to manage its software development life cycle (SDLC). The main components of their SDLC practices include:

- **Requirement Gathering:** The company collaborates closely with clients to gather detailed requirements, often using workshops, user stories, and functional specifications to ensure that the final product aligns with client needs.
- **Planning & Estimation:** Projects are divided into manageable phases, with sprints planned on a bi-weekly basis. Sprint planning involves setting clear goals and deliverables for each cycle, with regular check-ins to ensure progress is on track.
- **Design:** Coding Mountain's design process involves creating wireframes, mockups, and prototypes that are reviewed by stakeholders before the development phase begins.
- **Development:** Software is developed in iterations, with teams using agile methodologies to implement features incrementally. Continuous Integration (CI) practices are employed to ensure that the development process is smooth and efficient.
- **Testing:** Both manual and automated testing are used throughout the development process to ensure that the product meets the required quality standards. QA is involved at every stage, ensuring early detection of issues.

- **Deployment:** The company uses Continuous Deployment (CD) pipelines to ensure that the software is deployed quickly and reliably, enabling frequent releases and updates.
- **Maintenance:** After deployment, Coding Mountain provides ongoing support and maintenance, ensuring that the software continues to function effectively and receives timely updates.

4.4 Technological Infrastructure

Coding Mountain relies on modern and scalable technology infrastructure to support its projects and operations. Some of the key tools and technologies they use include:

- **Development Stack:** The company uses a diverse technology stack, including React, Node.js, Laravel, Django, JavaScript, and Python, depending on the nature of the project.
- **Cloud Infrastructure:** For hosting and scalable services, Coding Mountain utilizes cloud platforms such as AWS and DigitalOcean to ensure flexibility and scalability for client projects.
- **Project Management Tools:** Tools like Jira, Trello, and Slack are used for task management, communication, and team collaboration to ensure smooth workflow and project transparency.
- **Version Control:** Git, along with GitHub and GitLab, is used for version control and collaborative code development.
- **Security Measures:** The company maintains strong security protocols, including HTTPS encryption, secure authentication practices, and regular security audits to protect client data and ensure software safety.

CHAPTER 5: IMPLEMENTATION AND TESTING

This part of the documentation emphasizes on how the system was implemented, what open technologies were used, how the modules were implemented to create the whole system. It is the one of the most important chapters for this documentation as it highlights all the tools and technologies with detailed implementation instruction of this system. This part explains the coding structure, testing details, how the system can be accessed as well as the whole implementation architecture of this system.

5.1 Tools Used

CASE Tools

1. Jira

Jira is an industry-leading project management and issue tracking tool developed by Atlassian. It is primarily used for agile project management, bug tracking, and feature development. Our team utilized Jira for task assignment, sprint planning, and progress tracking throughout the development lifecycle. The tool provided valuable metrics on team velocity and helped identify potential bottlenecks in the development process.

5.2 Programming Languages and Technologies

1. Java

Java was used as the primary backend programming language due to its robustness, platform independence, and extensive library support. We implemented the core business logic, data processing algorithms, and service layer using Java 11. The Spring Boot framework was utilized to create RESTful APIs that communicate with the frontend application.

2. Angular

Angular framework was employed for developing the frontend components of the system. We utilized Angular's component-based architecture to create reusable UI elements, implemented reactive forms for data validation, and leveraged Angular services for managing state and API communication. The responsive design ensures the application works seamlessly across various devices and screen sizes.

3. **Node.js**

Node.js was used to create middleware services and build tools. It facilitated real-time communication features using Socket.IO for instant booking notifications and updates. Additionally, we utilized Node.js with Express to develop

CHAPTER 6: ANALYSIS AND FINDINGS

6.1 Current Scrum Methodology Analysis

Coding Mountain employs Scrum as its primary methodology for managing software development projects. Scrum's emphasis on iterative development, continuous improvement, and collaboration allows the company to deliver high-quality products while adapting to changing requirements and client needs.

At the core of Coding Mountain's approach is the Scrum framework, which divides work into small, manageable increments known as sprints. These sprints typically last two to four weeks and result in the delivery of a potentially shippable product increment. This iterative approach ensures that feedback from stakeholders and clients can be incorporated early and often, leading to continuous refinement and alignment with user needs.

Each Scrum team at Coding Mountain includes cross-functional members, including developers, designers, and testers, who collaborate closely to meet sprint goals. Daily stand-up meetings are held to ensure clear communication and to address any obstacles that may hinder progress. By following Scrum practices, the company maintains a high degree of transparency, accountability, and focus, ensuring that projects are completed on time and with high-quality results.

This methodology enables the team to quickly adapt to changes and deliver incremental value to clients, enhancing both efficiency and customer satisfaction.

6.2 Requirements Engineering Process Evaluation

The process of gathering and refining requirements at Coding Mountain is a collaborative effort that involves both clients and internal stakeholders. This process begins with comprehensive discussions to understand the project's objectives and user needs. The Scrum team works closely with clients to define the product backlog, a prioritized list of features, and functionalities that will be developed over time.

Coding Mountain uses agile practices in the requirements gathering phase, which ensures that requirements are continuously refined based on feedback from the stakeholders. This dynamic approach allows for flexibility as client needs evolve throughout the project lifecycle.

The team emphasizes clear documentation to ensure that all requirements are traceable and understood by all team members. Additionally, the team actively revisits the backlog during sprint planning sessions to adjust priorities based on the latest information and client feedback, ensuring that the development process remains aligned with user expectations.

6.3 Design and Development Practices

The design and development process at Coding Mountain is structured to support rapid delivery of working software. The Scrum teams employ a user-centric approach in their design practices, ensuring that the final product meets both functional and non-functional requirements. The team of UI/UX designers, developers, and quality assurance engineers collaborates closely throughout each sprint to create solutions that align with client goals.

Development is driven by frequent collaboration and iterative progress. Developers follow coding best practices, including the use of version control systems like Git, to ensure that the software is maintainable and scalable. Each sprint concludes with the delivery of a functional increment that is fully tested and ready for review by the client.

Coding Mountain also practices continuous integration and delivery, allowing for rapid deployment and regular updates. This approach enhances flexibility and enables the team to quickly respond to changing requirements or emerging opportunities.

6.4 Testing and Quality Assurance Procedures

Quality assurance (QA) is a crucial component of Coding Mountain's software development process. The company employs both manual and automated testing techniques to ensure that each product increment meets the highest standards of quality. QA engineers work closely with the development team to create comprehensive test plans and scripts that cover both functional and non-functional aspects of the software.

The Scrum teams prioritize early defect detection, using automated testing tools to run tests continuously and detect issues early in the development cycle. This approach ensures that issues are resolved promptly, and product quality is maintained throughout the sprints.

Additionally, bug tracking systems are used to document defects and track their resolution. QA engineers also perform regression testing to ensure that new features do not adversely affect the existing functionality of the product.

6.5 Project Management and Governance

Project management at Coding Mountain is largely governed by Scrum practices, where the Scrum Master plays a key role in facilitating the Scrum process and ensuring the team follows agile principles. The Scrum Master removes obstacles and ensures that the team stays focused on the sprint goals.

The company uses sprint reviews and retrospectives to promote continuous improvement. Sprint reviews allow the team to showcase the completed work and gather feedback from stakeholders, while retrospectives provide a space to reflect on the sprint process and identify opportunities for better collaboration and efficiency in future sprints.

Transparent communication, collaboration, and accountability are integral to the company's project management approach. Coding Mountain's project managers work closely with the Scrum teams to ensure that deadlines are met and that there is alignment with the overall project vision.

6.6 Key Challenges and Success Factors

Coding Mountain faces several challenges typical of software development projects, including managing evolving client needs and maintaining strong communication across cross-functional teams. However, the company's commitment to agile principles, continuous feedback, and collaboration has been instrumental in overcoming these challenges.

Key success factors for the company include a skilled and experienced workforce, a strong focus on user-centric design, and the ability to adapt quickly to changing market demands. The company's iterative approach to development allows for regular client feedback, helping to ensure that the final product meets client expectations and adds value to their business operations.

CHAPTER 7: DISCUSSION

7.1 Comparison with Industry Best Practices

In comparison to recognized industry standards, Coding Mountain demonstrates strong alignment with many key principles of Scrum. The company's focus on iterative development, frequent client interaction during sprint reviews, and a strong emphasis on collaboration between cross-functional teams is consistent with well-established Scrum practices. However, there are areas where Coding Mountain could further refine its processes to meet higher industry benchmarks.

For example, in many organizations following Scrum, the practice of documenting detailed sprint backlogs, well-defined user stories, and release plans is integral to effective project tracking. Coding Mountain could benefit from more formalized processes in these areas, especially in backlog refinement and detailed sprint planning. Additionally, while their focus on continuous delivery is commendable, expanding the use of automated testing and integrating more structured quality assurance practices could help them scale operations and maintain a high standard of quality.

7.2 Strengths of the Scrum Framework at Coding Mountain

Several strengths stand out in Coding Mountain's implementation of Scrum that contribute significantly to its success:

- **Agility and Flexibility:** Coding Mountain has effectively adapted Scrum's core principles to maintain flexibility while ensuring regular feedback loops. This flexibility allows the company to respond quickly to changing client needs or market conditions, which is a notable advantage in fast-paced environments.
- **Team Collaboration:** The company places a strong emphasis on teamwork, with daily stand-ups and cross-functional collaboration between developers, designers, and business analysts. This cohesion promotes rapid problem-solving and helps avoid delays due to miscommunication or misalignment of goals.
- **Client Involvement:** Regular client reviews and feedback during each sprint allow Coding Mountain to tailor solutions based on real-time input, ensuring that the final product meets client expectations and is aligned with evolving requirements. This

direct involvement enhances client satisfaction and minimizes the risk of delivering a product that does not meet their needs.

- **Timely Project Delivery:** Even with its resource constraints, Coding Mountain has maintained a track record of on-time project deliveries. Their disciplined adherence to Scrum ceremonies—such as sprint planning, reviews, and retrospectives—plays a vital role in keeping projects on track and ensuring that the team remains focused and productive.

7.3 Study Limitations

Several limitations should be taken into account when considering the findings of this study:

- **Single Case Study:** The study is based on a single case study of Coding Mountain, which may limit the generalizability of the results to other organizations or industries, especially those with different resources or project scopes.
- **Limited Data Access:** Some internal performance data and metrics, which could have provided deeper insights, were not available for analysis due to confidentiality constraints.
- **Time Constraints:** The research was conducted over a relatively short period, which restricted the ability to observe long-term improvements or conduct follow-up interviews to assess the impact of recommendations.

CHAPTER 8: CONCLUSION

8.1 Lessons Learnt

The lessons learned from analyzing Coding Mountain's use of Scrum methodologies emphasize the importance of flexibility and adaptability in software development. One of the key takeaways is that incorporating regular feedback loops with clients is essential to ensure the final product aligns with user needs and expectations. The iterative nature of Scrum allows for continuous refinement, helping teams stay agile and responsive to changes. Another valuable lesson is the significance of clear communication across cross-functional teams. Transparency in task allocation, progress tracking. Furthermore, the importance of automated testing was highlighted, as it helps improve the reliability and speed of development. Additionally, the need for a structured approach to knowledge sharing within the team was evident, as this can support better onboarding and maintain consistency across projects. By integrating these lessons into future practices, Coding Mountain can further streamline its development process, enhance team collaboration, and deliver even more high-quality, timely solutions to clients.

8.2 Conclusion

In conclusion, the analysis of Coding Mountain's software development practices reveals a strong alignment with industry best practices, particularly through their implementation of Scrum methodologies. The company's focus on iterative development, continuous feedback from clients, and close collaboration among cross-functional teams has enabled them to consistently deliver high-quality software solutions. However, there are still areas for improvement, such as refining documentation practices, expanding automated testing frameworks, and enhancing knowledge sharing within the team. By addressing these challenges and continuing to evolve their processes, Coding Mountain can enhance the maturity of their development practices, enabling them to scale and meet the increasing demands of the market. This case study underscores the importance of balancing agile flexibility with structured processes, and highlights how small to mid-sized companies can leverage Scrum to drive efficiency and deliver value to clients.

References:

- [1] CodingMountain, “About Us,” *CodingMountain.com*. [Online]. Available: <https://www.codingmountain.com/about>. [Accessed: May 14, 2025].
- [2] LinkedIn, “Coding Mountain,” *LinkedIn*. [Online]. Available: <https://np.linkedin.com/company/coding-mountain>. [Accessed: May 14, 2025].

Appendix- A

Questions About the Company

1. Can you describe the company?
2. What types of projects does your company do?
3. What are the primary goals of the IT department in this company?
4. What are the key responsibilities and expectations for this position?
5. What does a typical day look like for someone in this role?
6. What skills or experiences do you consider most important for success in this position?
7. What are your roles as a product manager? How do you overcome challenges as a product manager?
8. Can you share some examples of current or upcoming projects that the team is working on?
9. What development methodologies does your company employ (E.g., Agile, Waterfall)?
10. What are your project team's biggest challenges during each phase?
11. How does the team collaborate on projects, especially in a remote or hybrid work environment?
12. What tools and technologies are commonly used in your development process?
13. Can you share examples of successful client projects?
14. How does the company plan to adapt to changes in the software industry?
15. What are your short-term business goals (1-year)?
16. What are your long-term vision and goals (3-5 years)?
17. How many active clients do you currently have?
18. How do you gather and manage client or user requirements?
19. Can you share examples of successful client projects?
20. What programming languages are predominantly used?
21. What strategies do you use to keep your team motivated and productive?
22. What challenges have you faced on a software project, and how do you overcome them?
23. How do you estimate time cost and resources for software development projects?
24. Have you ever managed a project where the requirements kept changing? How do you manage it?

1. Can you describe the company?

Coding Mountain is a dynamic and expanding software development company with approximately 40 employees. The company is known for its international client base and works on delivering high-quality, customized software solutions. They serve clients from countries such as Netherlands, Israel, Japan, and United States. The company emphasizes innovation, adaptability, and client satisfaction as its core values.

2. What types of projects does your company do?

Coding Mountain primarily handles foreign projects ranging from enterprise-level web applications to subscription-based platforms and mobile applications. These projects are often developed for businesses seeking to modernize operations or launch digital products. The team focuses on problem-solving and delivering user-centric solutions.

3. What are the primary goals of the IT department in this company?

The IT department is dedicated to building robust and scalable software, ensuring seamless delivery, staying aligned with global tech trends, and maintaining high-quality standards in every phase of development. The department also supports collaborative efforts across teams and plays a critical role in implementing modern frameworks and tools.

4. What are the key responsibilities and expectations for this position?

As a Product Owner, responsibilities include conducting market research, gathering and analyzing client requirements, facilitating clear communication between teams, and translating business needs into actionable development tasks. The role also includes ensuring that the final product aligns with the customer's expectations and project goals.

5. What does a typical day look like for someone in this role?

Her day typically involves researching technologies and trends, meeting with clients and developers, organizing requirements, tracking project progress, and addressing any roadblocks faced by the team. The role demands active communication, quick decision-making, and attention to both strategic goals and technical details.

6. What skills or experiences do you consider most important for success in this position?

Essential skills include strong interpersonal and communication abilities, a keen eye for research and analysis, understanding of software lifecycles, adaptability to changing client demands, and the capacity to bridge gaps between technical and non-technical stakeholders.

7. What is your role as a project manager? How do you overcome challenges as a product manager?

Although she is currently a Product Owner, she aspires to become a Product Manager in the next 2–3 years. From her perspective, managing challenges, such as the handling remote team requires flexibility, problem-solving, and transparent stakeholder communication. She emphasizes teamwork, alternative exploration, and constructive discussion as her tools for overcoming such issues. Of course, the scrum event – Retrospective – helps her communicate such improvements with team openly.

8. Can you share some examples of current or upcoming projects that the team is working on?

While confidentiality prevents sharing project names, the team is engaged in developing and maintaining foreign web applications and subscription-based platforms. Many of these projects are customized to suit the regulatory and business environments of each country served.

9. What development methodologies does your company employ (e.g., Agile, Waterfall)?

Coding Mountain adopts scrum methodologies, promoting iterative progress, regular feedback, and flexibility. This allows the team to adapt quickly to evolving requirements and deliver incremental value to clients.

10. What are the biggest challenges your project teams face during each phase?

Time management is the biggest challenge faced by her as a product manager. Key challenges include managing unclear or shifting requirements and coordinating with clients

in different time zones. Technical feasibility and resource constraints also impact phases like design, integration, and testing.

11. How does the team collaborate on projects, especially in a remote or hybrid work environment?

Collaboration is maintained through structured communication channels involving project managers, joint managers, developers, and stakeholders. Regular meetings, shared documentation, and collaborative tools help maintain alignment and transparency. The tool Slack is used as collaboration tool.

12. What tools and technologies are commonly used in your development process?

Angular (TypeScript) is the primary front-end framework, while Java and Node.js is used on the backend. These tools support building robust and scalable web platforms that meet international standards and client-specific needs. Jira is used for their development process, as Jira can be used for testing as it allows for managing test cases, tracking test progress, and generating reports. It's a good option for teams already using Jira for project management and development. Jira helps unify processes for developers, testers, and QA teams on a single platform.

13. Can you share examples of successful client projects?

Though specific project names are confidential, Coding Mountain has delivered successful projects for clients in the Netherlands, Israel, United States, and Japan. These projects include business automation tools, mapping solutions, equity platforms and scalable platforms tailored to niche markets.

14. How does the company plan to adapt to changes in the software industry?

The company is proactive in staying updated with advancements in areas like AI. It promotes responsible usage of AI tools and educates team members on avoiding misuse, while exploring AI's potential to enhance productivity, automation, and intelligent features in client solutions. Moreover, the company encourages frequent knowledge sharing sessions among the team and colleagues. Apart from it, the company also hosts the

Spotlight sessions once a month where 3 speakers (randomly selected among the employees) present or speak about something interesting or informative to boost the public speaking confidence.

15. What are your short-term business goals (1 year)?

In the short term, the company aims to expand its international client base, improve internal team collaboration, and streamline project delivery. There's also a focus on process optimization and exploring new technologies.

16. What are your long-term vision and goals (3–5 years)?

Her personal goal is to transition into a Product Manager role. On a company level, the long-term vision includes scaling the business, entering new global markets, adopting emerging technologies like AI more deeply, and becoming a trusted partner for clients worldwide.

17. How many active clients do you currently have?

While an exact number wasn't provided, Coding Mountain has several active clients, mainly from Asian countries. The team works on long-term engagements and values sustained client relationships.

18. How do you gather and manage client or user requirements?

Requirements are collected through detailed conversations with clients and clarified through continuous stakeholder collaboration. Emphasis is placed on mutual understanding and regular communication to reduce ambiguity and align development goals. The market and competitive research also help with the user requirements. Also, the backlog refinement session (an official scrum activity) clarifies the user requirement among the scrum team.

19. What programming languages are predominantly used?

Angular, Java, and Node.js are the core technologies used. Angular is chosen for responsive, interactive front-end development, while Java and Node.js supports robust, secure backend development.

20. What strategies do you use to keep your team motivated and productive?

To maintain team morale and engagement, the company fosters a friendly environment where members share updates, collaborate openly, and occasionally enjoy team outings or entertainment activities. This strengthens bonds and improves productivity. Moreover, the team also shares their adventurous pictures so that the team can get to know each other better and keep the interesting conversation ongoing apart from just monotonous works.

21. What challenges have you faced on software projects and how do you overcome them?

One major challenge was complex design of the equity management platform, which lacked intuitive designs. The team overcame this through collaborative research, exploring alternatives, and maintaining close communication with the stakeholders and design team to realign expectations.

22. How do you estimate time, cost, and resources for software development projects?

Estimations are made collaboratively by the joint manager, project manager, stakeholders, and the product owner. The process relies on experience, open communication, and consensus, which ensures practical and realistic planning.

23. Have you ever managed a project where the requirements kept changing? How do you manage it?

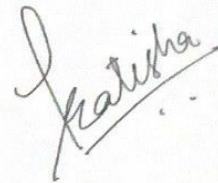
Yes, and the product owner manages such changes by ensuring flexible planning, ongoing communication with the client, and aligning the development team to pivot direction when necessary. Agile methodology supports handling changing requirements effectively and keeping the project on track.

Stamp



.....
Coding Mountain
Lagankhel, Lalitpur

Signature

A handwritten signature in black ink, appearing to read "Pratisha".

.....
Pratisha Shakya
Product Owner
Coding Mountain