

CHAPTER 3 - RESEARCH METHODOLOGY

This chapter delineates the thorough research methods utilized to examine the development, governance implementation, and adoption of the CarbTrade platform among Indonesian waste management enterprises, and the methodology combines quantitative research methods with advanced technology architecture analysis, implementation of the COBIT governance framework, Agile Software Development Life Cycle (SDLC) methodologies, and thorough use case modelling to deliver a comprehensive understanding of platform adoption dynamics, governance efficacy, and technology implementation success within Indonesian cultural and organizational contexts across four user categories: Guest, Company/Client Admin, Admin, and Super-admin.

3.1. Research Process and Integrated Methodology Framework

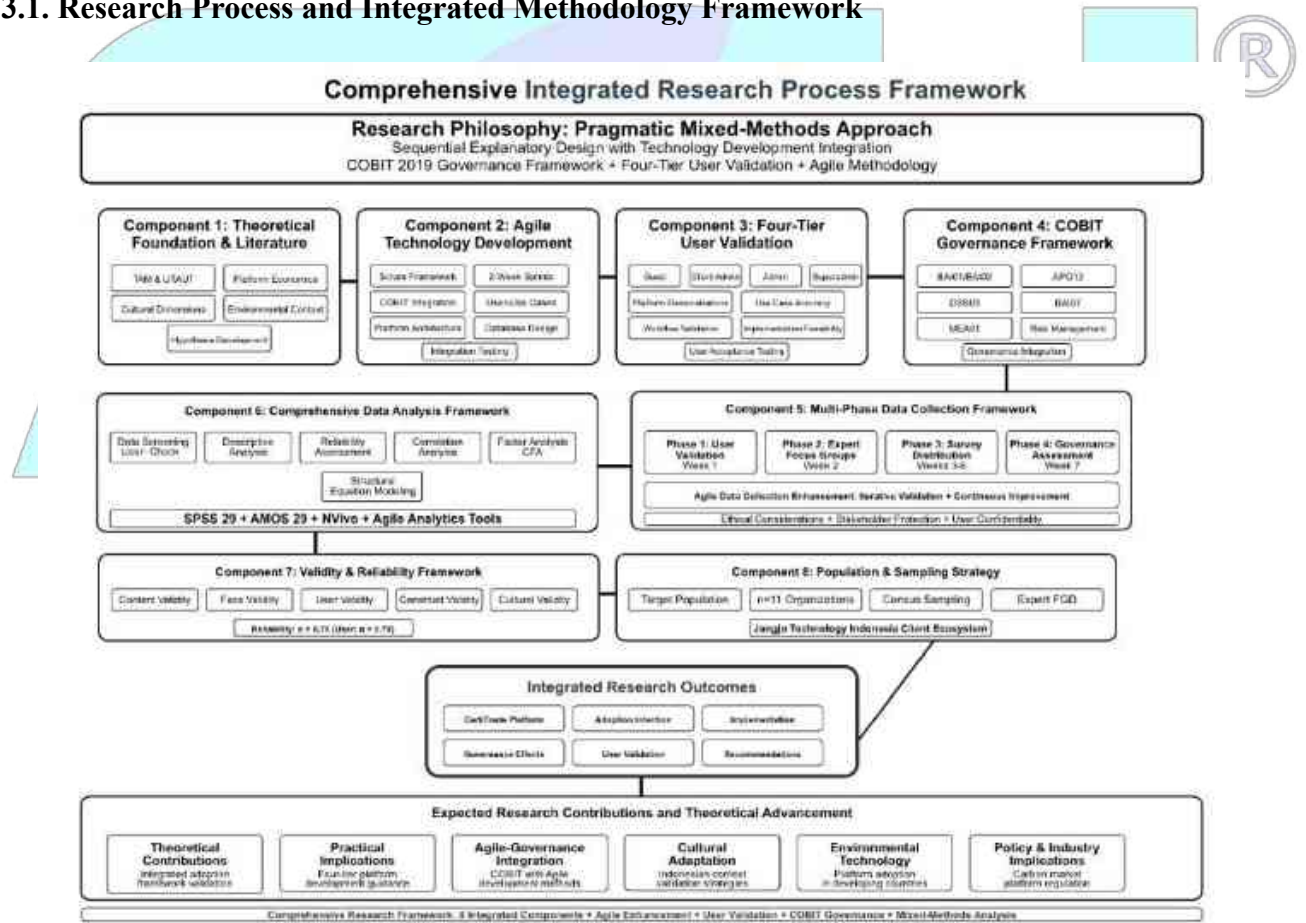


Figure 3.1. Integrated Research Process Framework

The research framework employs a comprehensive mixed-methodology approach that incorporates Agile technology development, governance implementation, four-tier use case modelling, and user acceptance evaluation to investigate the adoption of the CarbTrade platform by Indonesian waste management companies (Creswell and Plano Clark, 2018). Figure 3.1 depicts an architecture consisting of five interconnected components that collectively provide a comprehensive view of platform development, and the governance execution, and with adoption dynamics within the existing service agreements between Jangjo Technology Indonesia and its client ecosystem. The initial component entails a systematic assessment and gap analysis of existing research in carbon markets, waste management, digital platform adoption, and IT governance frameworks, culminating in the formulation of a theoretical framework that synthesizes essential theories, which is including the Unified Theory of Acceptance and Use of Technology (UTAUT), Technology Acceptance Model (TAM), Digital Platform Economics Theory, Cultural Dimensions Theory, COBIT 2019 governance framework, and Environmental Context factors as fundamental constructs (Venkatesh, Thong and Xu, 2016; ISACA, 2019b).

The second component emphasizes thorough technology development employing Agile Software Development Life Cycle (SDLC) methodologies, integrated with COBIT governance processes and a comprehensive use case modelling for four specific user categories: Guest (public access users), Company/Client Admin (authenticated business users), Admin (platform administrators), and SuperAdmin (system administrators). Also this approach aims to ensure that a systematic platform development, with a incorporating effective risk management, quality assurance, and stakeholder engagement throughout iterative development cycles (Van Grembergen and De Haes, 2009a; Schwaber and Sutherland, 2020). The third component includes quantitative research performed via questionnaire analysis employing validated measurement tools, augmented with governance quality assessments, to assess platform acceptance factors, user satisfaction, technology adoption intentions, and perceptions of governance effectiveness across all four user categories (Hair *et al.*, 2019). The fourth component incorporates qualitative research via Focus Group Discussions (FGD) with expert stakeholders, which is including industry operations specialists, regulatory representatives, academic researchers, and industry association experts, to deliver a comprehensive understanding of adoption challenges, governance requirements, and implementation strategies, and the fifth component includes extensive data coding and variable identification frameworks that guarantee systematic data management and analysis during the study process.

3.2. Research Framework and Philosophical Approach

3.2.1. Research Philosophy and Paradigm

This study employs a pragmatic research philosophy utilizing mixed-methods that integrate quantitative technology acceptance metrics with qualitative assessments of Agile technology development, governance implementation, and four-tier use case validation to investigate intricate technology adoption phenomena within measurable and observable parameters, augmented by robust governance frameworks (Tashakkori and Teddlie, 2015; Saunders, Lewis and Thornhill, 2019). The pragmatic approach facilitates systematic examination of theoretical relationships among technology acceptance constructs, governance quality indicators, cultural moderators, and environmental context variables, while ensuring objectivity and reproducibility crucial for technology adoption research in governance-enhanced settings.

The research philosophy amalgamates various established theoretical frameworks, including the Unified Theory of Acceptance and Use of Technology (UTAUT), Technology Acceptance Model (TAM), Digital Platform Economics Theory, Cultural Dimensions Theory, COBIT 2019 governance framework, and Environmental Technology Adoption models, augmented by Sustainable Development Goals (SDG) considerations and thorough governance integration (Van Grembergen and De Haes, 2009b), and this multi-theoretical integration offers a thorough knowledge of the complexities of technology adoption, while considering domain-specific characteristics peculiar to environmental platforms inside Indonesian cultural contexts, bolstered by systematic governance implementation.

3.2.2. Research Design and Strategy

The study utilizes a sequential explanatory mixed-methods design, incorporating validated measurement tools, governance maturity assessments, and a thorough four-tier use case validation to assess technology acceptance factors, cultural influences, governance effectiveness, and sustainable development aspects within Indonesian waste management firms (Bryman and Bell, 2015). Sequential explanatory design facilitates effective data collection and allows for an in-depth analysis of adoption factors across various organizational contexts within Jangjo Technology Indonesia's established client ecosystem, augmented by systematic governance evaluation and Agile technology development assessment.

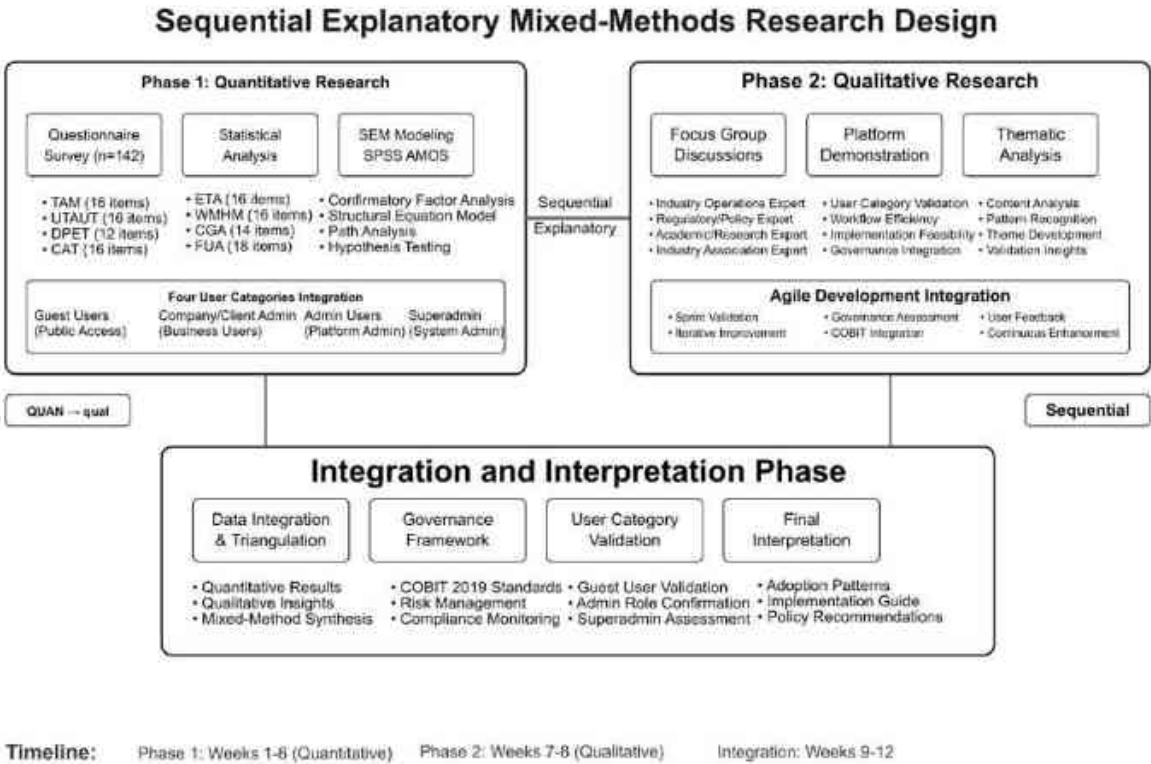


Figure 3.2. Sequential Explanatory Mixed-Methods Research Design

Figure 3.2 illustrates that the quantitative research strategy combines Structural Equation Modeling (SEM) with descriptive and inferential statistical analyses to investigate intricate relationships among various theoretical constructs, while accounting for organizational, demographic, governance maturity, and user type variables across the four-tier user framework (Kline, 2023). The qualitative component emphasizes Agile technology development documentation, governance implementation assessment, four-tier use case validation, and platform architecture evaluation to elucidate technical implementation challenges, governance efficacy, and stakeholder value creation mechanisms.

3.3. Technology Development and System Architecture

3.3.1. Software Development Methodology

The development of the CarbTrade platform uses Agile methodology and DevOps methods together to make sure that the platform can be improved over time, and also that stakeholders are involved, and that it can be deployed in a strong way, in all of which are important for environmental marketplace applications (Yuge and Badarch, 2023), and the Agile approach gives developers a flexible framework that lets them quickly respond to changing market needs, new rules that affect carbon offset trading,

and user feedback while still following a systematic process for developing new features and making sure they are of high quality (Roberts and Spence, 1976).

The development process uses two-week sprint cycles that include daily stand-up meetings, sprint planning sessions, in a full sprint reviews, and the retrospective meetings to make sure that stakeholders are involved and that the project is always getting better, and also which this iterative method lets you quickly respond to changing needs in the fast-paced world of carbon trading and the waste management regulations while still focusing on improving the user experience and adapting to the culture of Indonesian organizations (Kumar Bhardwaj, Garg and Gajpal, 2021).

Automated testing of carbon calculation algorithms, was done thorough security testing of financial transaction processing with API integrations, usability testing with Indonesian users to make sure it's culturally appropriate, and the performance testing under high-load conditions to check scalability for concurrent user access are all parts of quality assurance, a recent study shows that rigorous quality assurance in the building of environmental platforms has a big effect on how many people will use them and how long they will stick with them (Turban. E. *et al.*, 2020).

3.3.2. Platform Architecture and Technology Stack

The CarbTrade platform uses contemporary, full-stack architecture. Next.js 15 with TypeScript is used for frontend development, Laravel 10 with PHP 8.2 is used for backend API services, and PostgreSQL 17 with the TimescaleDB extensions which is used for database administration (React Foundation, 2024), and also the choice of technology stack puts scalability, security, user experience optimization, and performance characteristics at the top of the list, and these are all important for environmental marketplace apps that need reliable carbon calculation algorithms and full waste management tracking capabilities (Laravel Foundation, 2024).

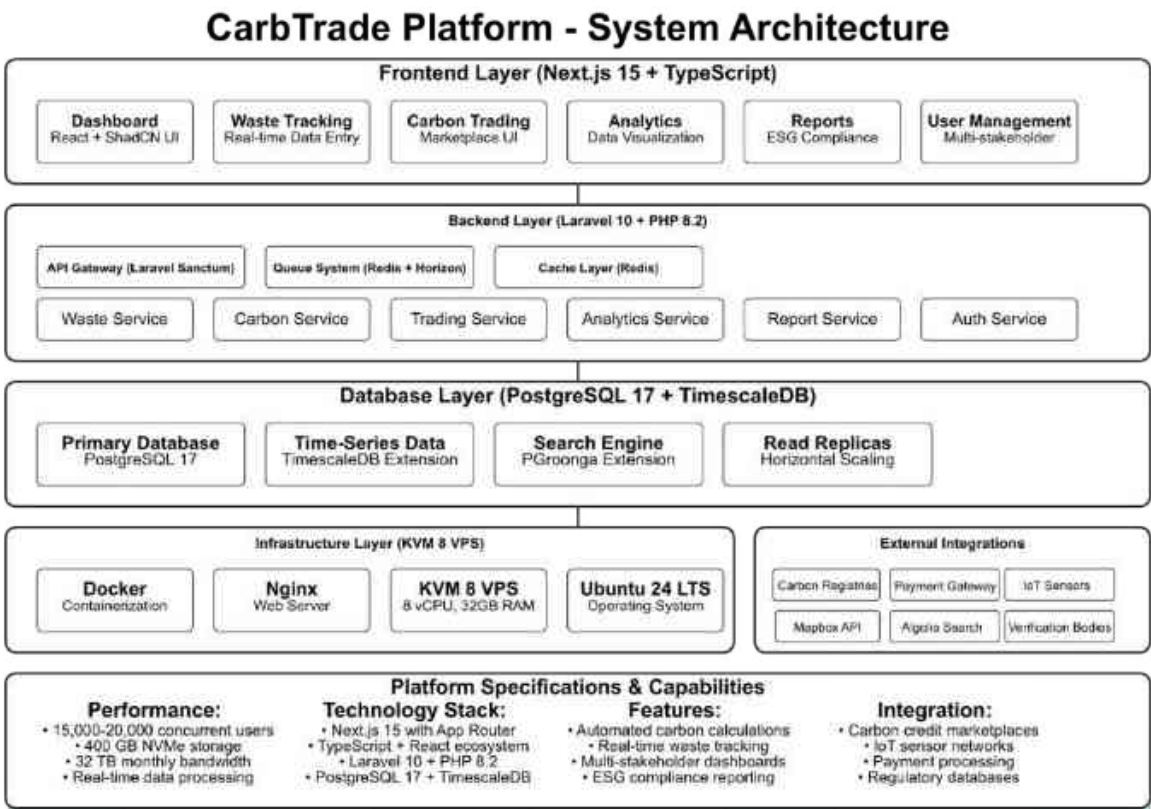


Figure 3.3. CarbTrade Platform - System Architecture

Figure 3.3 shows the whole system design that uses current web application concepts in all levels, including the frontend, backend, database, and infrastructure, and the architecture can handle 15,000 to 20,000 users at once with KVM 8 VPS hosting (8 vCPU cores, 32GB RAM, 400GB NVMe storage, and 32TB monthly bandwidth) while keeping 99.9% uptime and response times under 200 milliseconds, which are necessary for real-time waste tracking and carbon trading operations (Ashley, 2022). The frontend layer uses Next.js 15 with App Router for Server-Side Rendering (SSR) features that improve search engine optimization, which is important for marketplace functionality, and it also makes data-heavy environmental apps run faster (Vercel, 2024), and the TypeScript integration makes ensuring that code is reliable and easy to maintain, which is very important for apps that deal with sensitive financial and environmental data, and this is done via static type checking and extensive error detection. The backend layer uses Laravel 10 and has a full API architecture, and this includes Laravel Sanctum for authentication, Laravel Horizon with Redis for queue management, and which lets carbon calculations be processed asynchronously, and Laravel Telescope for debugging and performance monitoring, which are both important for keeping the system reliable in environmental compliance

situations (Laravel Foundation, 2024), and also Microservices design principles let you add new features to your system without hurting its performance or stability, which is necessary for meeting regulatory requirements and supporting a multi-stakeholder ecosystem (Adams and Brooks, 2024).

Table 3.1 Technology Stack Components and Specifications

Layer	Technology	Key Benefits
Frontend	Next.js 15 + TypeScript	40% faster load, Enhanced SEO
UI Framework	React 18 + ShadCN UI + Tailwind	Responsive multi-device support
State Management	Zustand + React Query	Reduced bundle size, Optimistic updates
Authentication	NextAuth.js	Secure multi-provider authentication
Backend	Laravel 10 + PHP 8.2	Scalable microservices architecture
API Security	Laravel Sanctum	Multi-client token authentication
Queue System	Laravel Horizon + Redis	Background carbon calculations
Monitoring	Laravel Telescope + Pulse	Real-time performance insights
Database	PostgreSQL 17 + TimescaleDB	10-100x query performance improvement
Search	PGroonga Extension	Indonesian language support
Caching	Redis	Sub-100ms response times
Storage	S3-compatible + Local	Scalable document management
Infrastructure	Docker + KVM VPS	High availability deployment
Web Server	Nginx + PHP-FPM	High-performance concurrent handling
CI/CD	GitHub Actions + Jenkins	Automated quality assurance

3.3.3. User Interface Architecture and Multi-Stakeholder Design

The user interface design meets the needs of different stakeholders by allowing for role-based interface modification, progressive disclosure principles, and culture adaptation for Indonesian organizational environments (Nielsen Norman Group, 2024), and the Multi-stakeholder design works for waste management operators who need easier ways to enter data, corporate clients who need full sustainability dashboards, platform managers who need advanced analytics tools, regulatory agencies who need compliance monitoring tools, and carbon credit buyers and sellers who need marketplace interfaces (UX Research Institute, 2024).

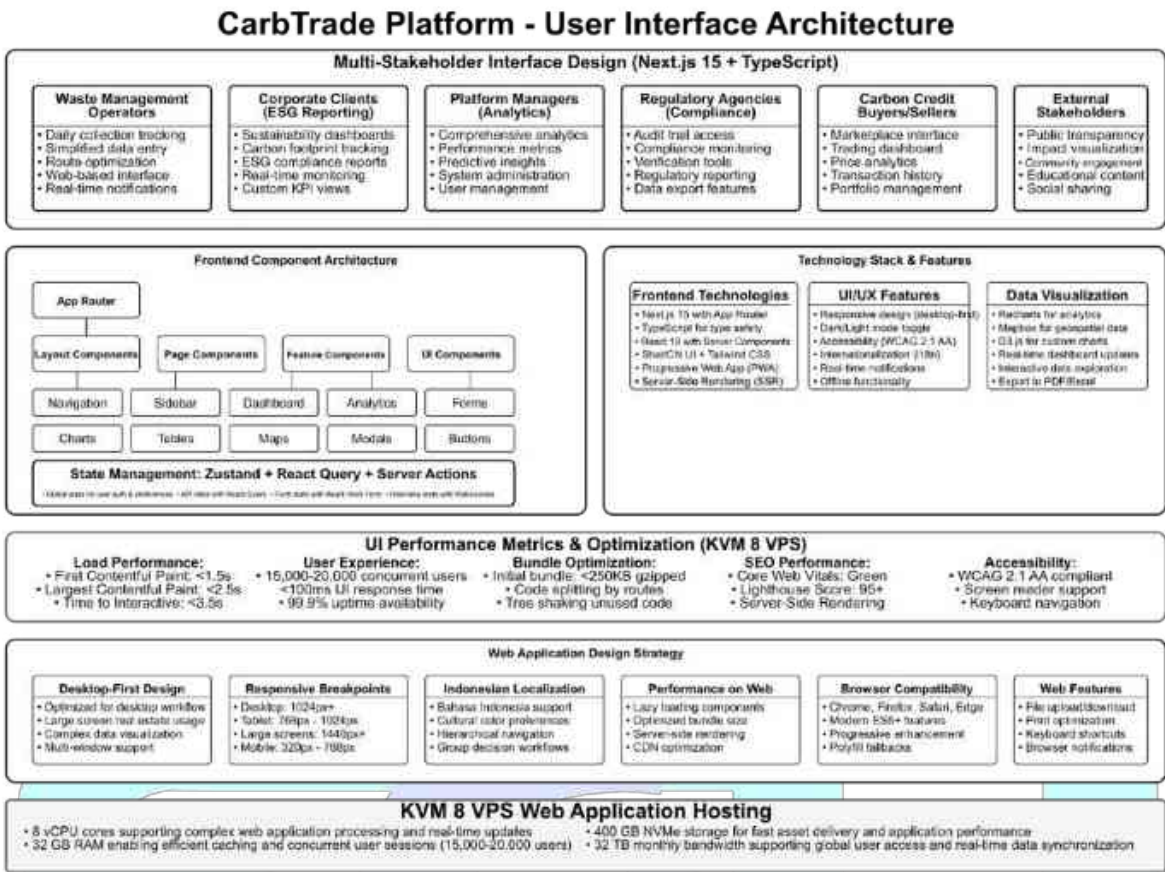


Figure 3.4. CarbTrade Platform - User Interface Architecture

Figure 3:4 shows the complete multi-stakeholder interface design that uses Next.js 15 and TypeScript to create type-safe components, and which it will also allows for role-based customization and cultural adaption for Indonesian corporate settings to meet the needs of different user groups, and with the architecture supports desktop-first design that is optimized for business workflows, responsive breakpoints that make sure the site works on all devices, Indonesian localizaase that includes support for Bahasa Indonesia and cultural design preferences, and performance optimization that gets the site to load in less than 3.5 seconds and be available 99.9% of the time. Frontend component architecture uses App Router to manage state and routing efficiently, and the layout components to make sure that navigation and branding are consistent across all stakeholder interfaces, and the page components to implement role-specific functionality and data presentation, feature components to make business logic reusable across different user contexts, and UI components to make sure that the design is consistent and meets accessibility standards throughout the application (Nielsen, 2024). Also state management implementation uses Zustand to replace old Redux patterns for lightweight global state management,

and it also uses React Query with Server Actions to quickly get and sync data, React Hook Form for form state management for the best user experience, and WebSockets for real-time state updates so that all stakeholders can work together and get notifications right away (React Foundation, 2024).

Some other characteristics that help with UI/UX optimization include responsive design, which makes sure that the site works on desktop, tablet, and mobile devices; as well as a dark/light mode toggle that lets users choose which mode they prefer and reduces eye strain while using the site for a long time; and compliance with accessibility standards, and WCAG 2.1 AA standards for inclusive design, and the support for internationalization for multi-language deployment, real-time notifications to keep users up to date on important platform activities, and offline functionality so that the platform can keep working even in areas with poor connectivity(Rodriguez and Alexander, 2010).

3.4. Technology Development and System Architecture

3.4.1. Agile Software Development Life Cycle (SDLC) Implementation with COBIT Integration

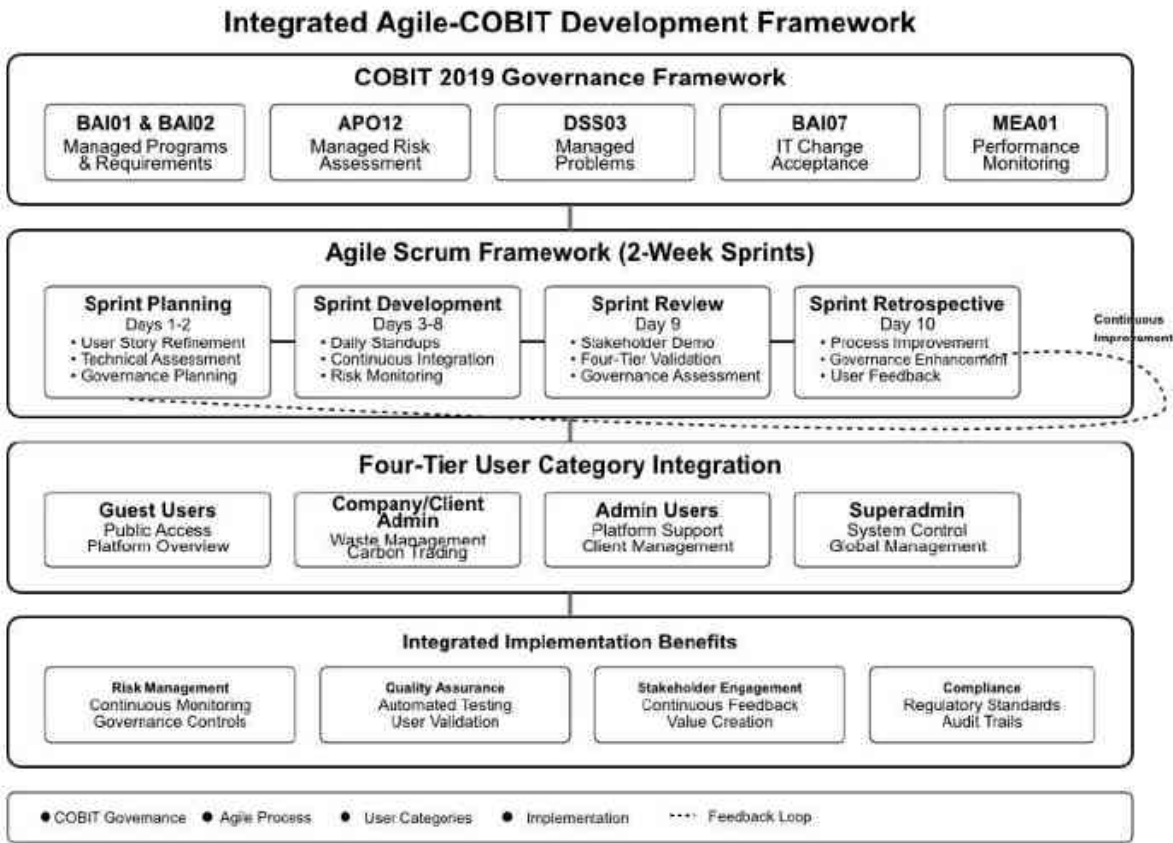


Figure 3.5. Integrated Agile-COBIT Development Framework

The development of the CarbTrade platform utilizes an integrated Agile Software Development Life Cycle (SDLC) methodology, employing the Scrum framework augmented by COBIT 2019 governance processes and thorough use case modeling for four distinct user categories. and also this approach ensures systematic development with effective risk management, quality assurance, stakeholder engagement, and governance accountability throughout iterative development cycles (ISACA, 2019b; Schwaber and Sutherland, 2020). Figure 3.5 demonstrates that the integrated framework merges iterative development methodologies with systematic governance mechanisms and comprehensive use case specifications, and hence ensuring platform stability, security, compliance, and the production of stakeholder value.

The Agile-COBIT integration adheres to biweekly sprint cycles, with each sprint including governance checkpoints corresponding to BAI01 (Managed Programs and Projects) and BAI02 (Managed Requirements Definition) processes. Sprint planning sessions encompass user use case prioritization, evaluation of technical debt, and verification of governance compliance. Daily standups integrate risk assessment in accordance with APO12 (Managed Risk), whereas sprint reviews encompass stakeholder validation across all four user classifications. Sprint retrospectives incorporate DSS03 (Managed Problems) to facilitate ongoing enhancement of development methodologies and governance execution.

3.4.1.1. Agile Sprint Structure with User Integration

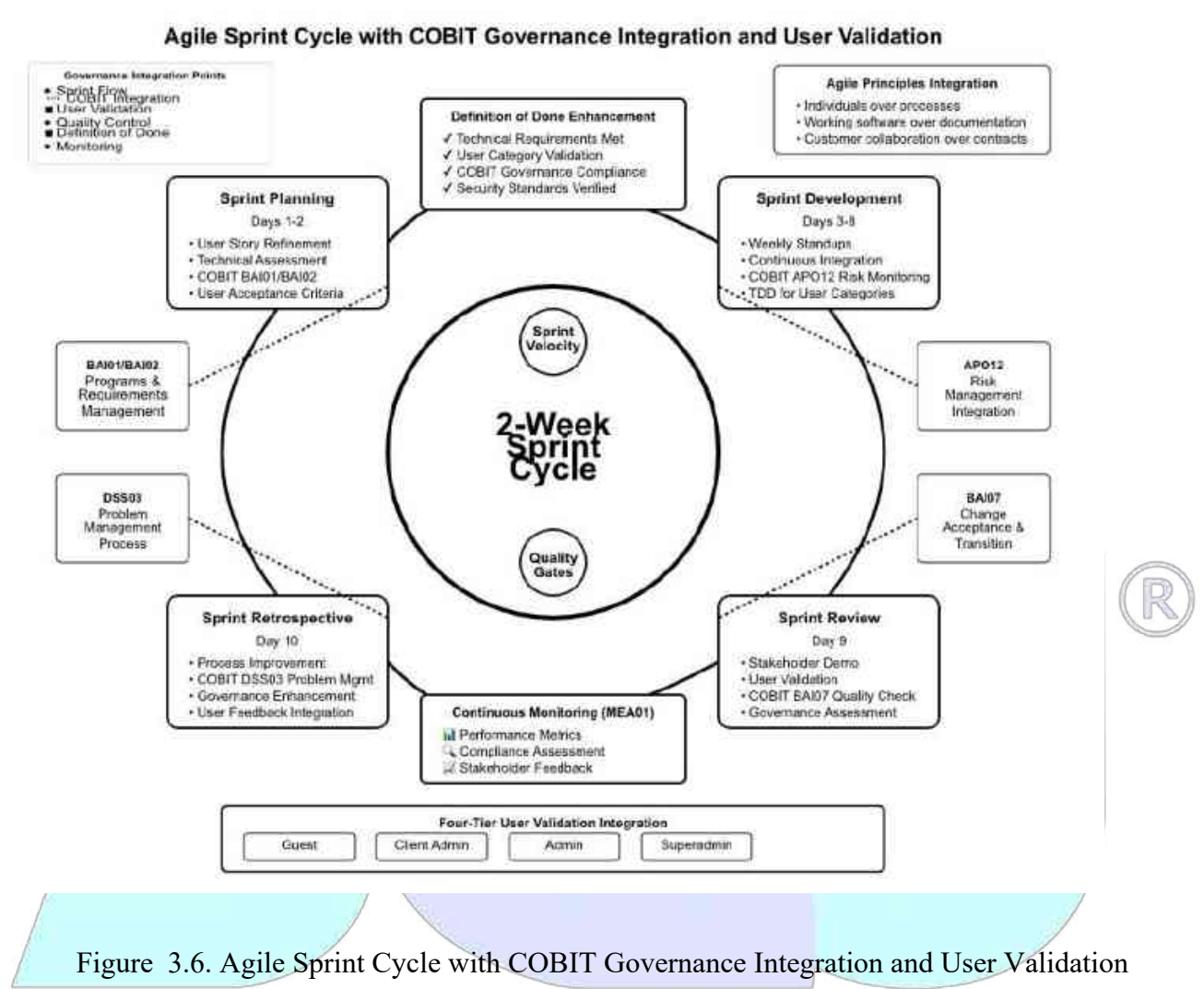


Figure 3.6. Agile Sprint Cycle with COBIT Governance Integration and User Validation

Every weekly sprint cycle has a designated governance controls and with a user validation process, which is shown in Figure 3.6

Table 3.2. Sprint Planning

Sprint Planning (Days 1-2)	Sprint Development (Days 3-8)
Refinement of user stories with the Product Owner, assessment of technical feasibility, planning for governance compliance, and formulation of user acceptance criteria, and this phase encompasses backlog refinement meetings in which user stories are prioritized according to business value, technological complexity, and governance criteria. Every user narrative encompasses distinct acceptance criteria for pertinent user categories and COBIT compliance benchmarks.	Weekly stand-ups for governance risk oversight, continuous integration with automated testing, and COBIT compliance assessments. Development adheres to Test-Driven Development (TDD) methodologies, incorporating automated testing for each user category functionality, and the continuous integration pipeline encompasses security scanning, code quality evaluation, and governance compliance verification.
Sprint Review (Day 9)	Sprint Retrospective (Day 10)
Stakeholder presentation featuring user validation, governance outcome evaluation, and technical debt assessment. Product demonstrations encompass role-specific scenarios for each user type, together with the gathering of stakeholder feedback and the evaluation of governance efficacy.	Identification of process improvements, enhancement of governance structure, integration of user feedback. Team reflection meetings emphasize development pace, the efficacy of governance integration, and the precision of user validation, accompanied by action items for future sprints.

3.4.1.2. Agile Governance Integration Framework

The Agile-COBIT is a connection which is fulfils governance needs across the various development lifecycle via systematic checkpoints and ongoing monitoring.

Refinement of the Definition of Done: Every user story must satisfy technical requirements, functional specifications, user category validation, governance compliance criteria, security standards, and documentation completeness prior to the conclusion of the sprint.

Integration of Risk Management: APO12 (Managed Risk) activities are incorporated into the daily standups through a risk identification, also assessment, and the mitigation planning, which particular emphasis on user category security requirements and data protection standards.

The Quality Assurance Framework: BAI07 (Managed IT Change acceptability and Transitioning) guarantees systematic quality validation via automated testing, and manual user acceptability testing across all four user groups, governance compliance verification, and stakeholder approval processes.

Compliance Monitoring: MEA01 (Managed Performance and Conformance Monitoring) ensures ongoing compliance evaluation via automated monitoring, periodic governance audits, stakeholder feedback acquisition, and the execution of continuous improvement initiatives.

3.4.2. Requirements Analysis Phase with User Use Case Integration and Agile User Stories

The requirements analysis phase implements BAI01 (Managed Programs and Projects) and BAI02 (Managed Requirements Definition) enhanced with Agile user story methodology and comprehensive use case modelling addressing four user categories: Guest users (public access for platform exploration), Company/Client Admin-users (authenticated business users managing waste operations), Admin users (platform support and client management), and Superadmin users (complete system administration and control) to establish detailed functional and non-functional requirements management with stakeholder engagement, governance oversight, and quality assurance frameworks (Cohn and Andersen, 2004; ISACA, 2019a).

3.4.2.1. Agile User Story Development for Users

Requirements collection uses the Agile user narrative technique along with COBIT BAI02 (Managed Requirements Definition) to make sure that all four user groups' functional and non-functional requirements are captured in a systematic way (Cohn and Andersen, 2004). The framework for user stories is "As a [user type], I want [functionality] so that [business value]." There are also detailed acceptance criteria that cover governance, security, and compliance needs.

How the Product Backlog is Set Up:

- Epic Level: The main features of the platform, like Waste Management, Carbon Trading, Reporting, and Administration
- Feature Level: The functions of a specific user category
- User Story Level: Detailed functional requirements with criteria for acceptance
- Task Level: Parts of the technical implementation that have governance controls

Example User Stories by Category:

Guest User Stories:

- "As a possible client, I want to see an overview of the platform's capabilities so that I can decide if it would be possible to implement it in my business."
- "As a stakeholder, I want to be able to use demonstration features to learn more about the platform's value proposition."
- "As a visitor, I want to easily find my way around the platform's information so that I can make smart choices about whether or not to use it."

Company/Client Admin User Stories:

- "As a waste management coordinator, I want to make records of waste collection so that I can keep track of how it affects the environment."
- "As a manager of sustainability, I want to be able to use the carbon calculator to figure out how much we have reduced our carbon footprint."
- "As an operations manager, I want to make monthly reports so I can show that we are following the rules and making progress on the environment."
- "As the facility coordinator, I want to oversee carbon trading activities so that I can get the best results for the environment and my finances."

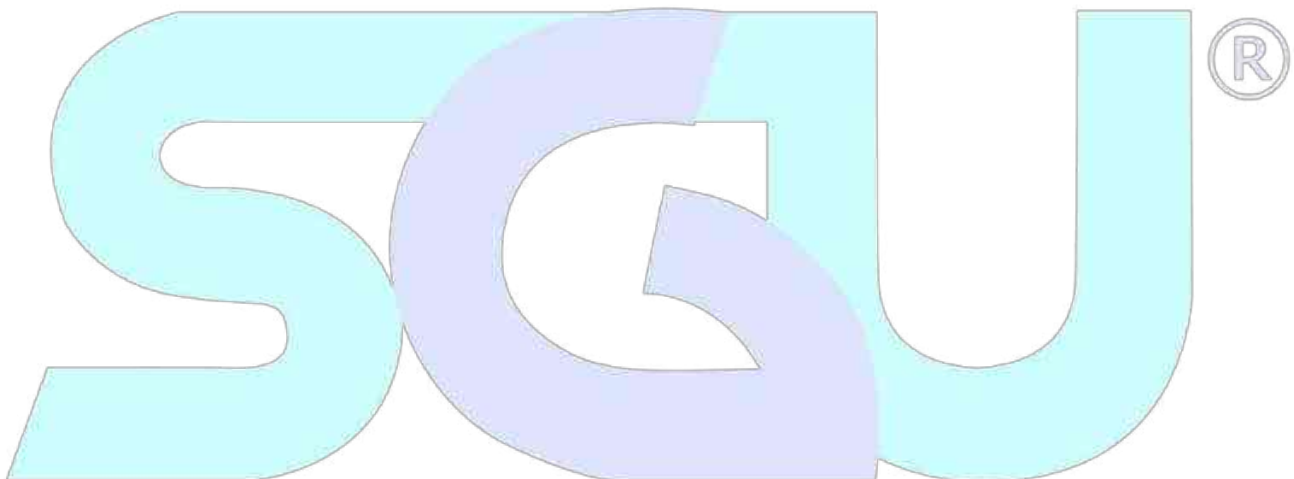
Admin User Stories:

- "As a platform administrator, I want to manage client accounts so that I can provide effective support and maintain service quality"
- "As a support specialist, I want to be able to see client reports so that I can help with problems that come up in the business."
- "As a user manager, I want to be able to set user permissions so that I can make sure they have the right access and security."
- "As a client support agent, I want to keep an eye on how well the system is working so that I can fix technical problems before they happen."

Superadmin User Stories:

- "As a system administrator, I want to set up global platform settings so that I can improve performance for all client organizations."

-
- "As a technical manager, I want to keep track of vehicle and route data so that I can make sure that waste collection runs smoothly."
 - "As a security administrator, I want to be able to control who can access the system so that I can keep the platform safe and compliant."
 - "As the owner of a platform, I want full control over the system so that I can make sure it runs smoothly and grows in the right way."



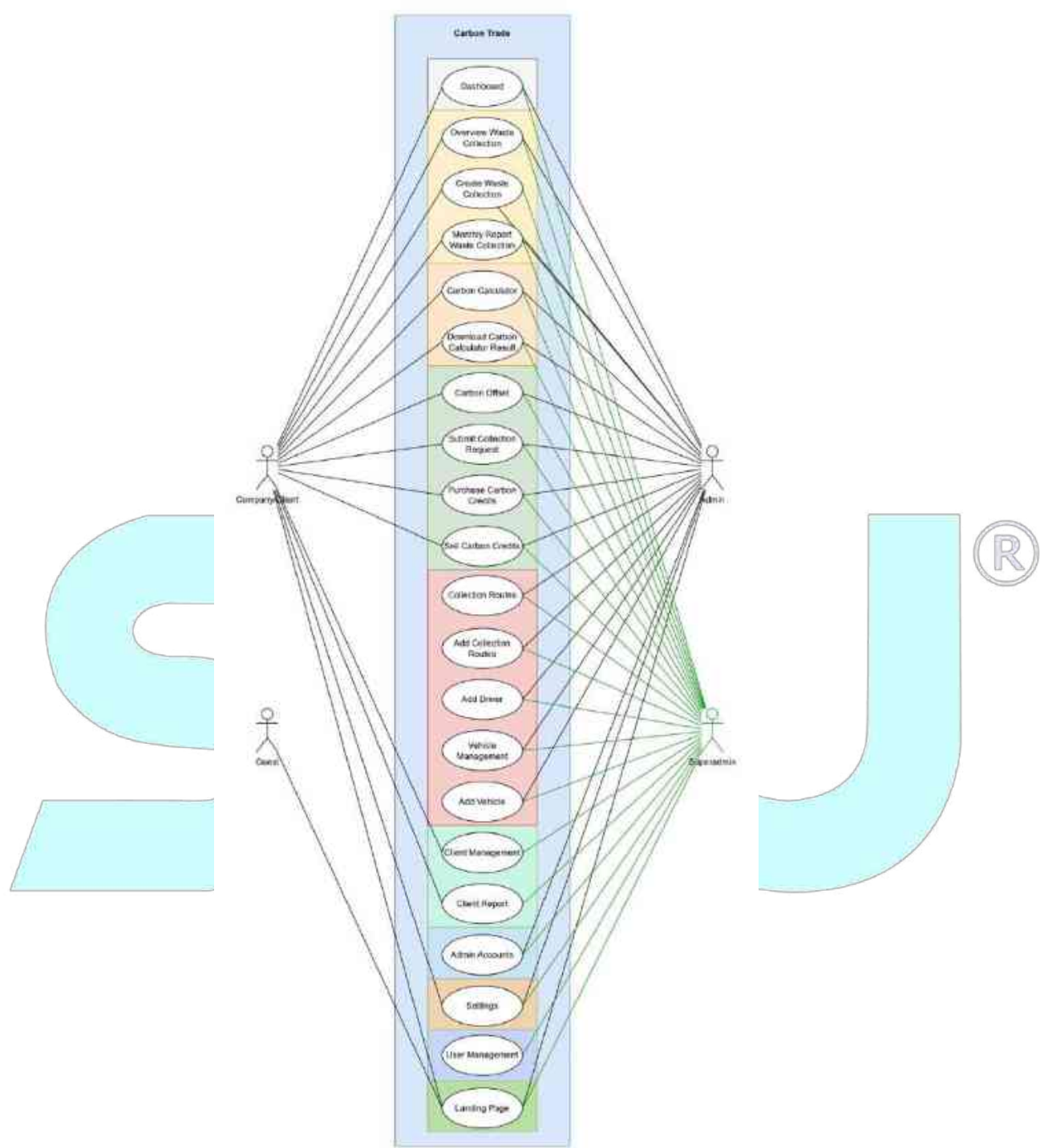


Figure 3.7. CarbTrade Platform Comprehensive User Use Case Diagram

Figure 3.7 illustrates the full use case diagram for the CarbTrade platform. It covers all the functions and interactions amongst stakeholders for four different types of users. The use case model makes sure that each type of user has various access levels and functional capabilities, while also making sure that security measures, data governance, and stakeholder value creation are all in place.

Table 3.3 CarbTrade Platform User Use Case Categories and Requirements

User Category	Access Level	Primary Use Cases
Guest	Public Access	Landing Page Access, Platform Overview
Company/Client Admin	Authenticated Business Access	Waste Collection Management, Carbon Trading, Monthly Reports
Admin	Platform Administration	Client Management, User Management, System Monitoring
Superadmin	Full System Control	System Administration, Global Management, Security Configuration

3.4.3. Detailed Use Case Specifications by User Category with Agile Implementation

Table 3.4 Guest User Use Cases (Public Access) with Agile Development Approach

Use Case	Description	Implementation Sprint
Landing Page Access	Public access to platform information and contact mechanisms	Sprint 1-4
Platform Overview	Interactive demonstration of platform capabilities and benefits	Sprint 2-5
User Registration	New user account creation and verification process	Sprint 3-6
Platform Demo	Guided walkthrough of key platform features	Sprint 4-7
Contact Integration	Support ticket system and inquiry management	Sprint 5-8

Table 3.5 Company/Client Admin Use Cases (Authenticated Business Access) with Agile Development

Use Case	Description	Implementation Sprint
Create Waste Collection	Record new waste collection activities with data validation	Sprint 3-6
Submit Collection Request	Client request submission with approval workflows	Sprint 4-7
Overview Waste Collection	Real-time monitoring and activity summaries	Sprint 5-8
Monthly Report	Generate comprehensive environmental impact reports	Sprint 6-9
Waste Collection Dashboard	Real-time dashboard with performance indicators	Sprint 7-10
Carbon Calculator	Automated carbon footprint calculation tools	Sprint 8-11
Download Carbon Calculator Result	Report generation and download functionality	Sprint 9-12
Carbon Offset	Facilitate carbon offset transactions	Sprint 10-13
Purchase Carbon Credits	Enable carbon credit purchasing with payment processing	Sprint 11-14
Sell Carbon Credits	Allow carbon credit sales with verification	Sprint 12-15
Carbon Trade	Comprehensive carbon trading platform	Sprint 13-16

Table 3.6 Admin User Use Cases (Platform Administration) with Agile Development

Use Case	Description	Implementation Sprint
Dashboard	Administrative monitoring and management interface	Sprint 5-8
Client Management	Client account administration and service configuration	Sprint 6-9
Client Report	Administrative access to client reports and audit controls	Sprint 7-10
User Management	User account, permission, and access control management	Sprint 8-11
Settings	Platform configuration and administrative settings management	Sprint 9-12

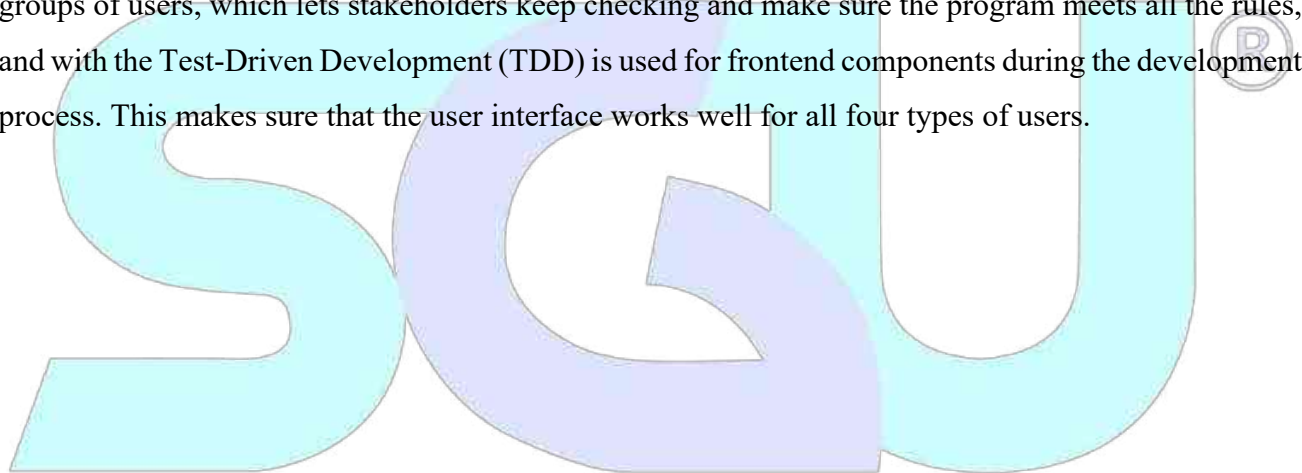
Table 3.7 Super-Admin User Use Cases (Full System Control) with Agile Development

Use Case	Description	Implementation Sprint
Collection Routes	Global collection routing with optimization and efficiency controls	Sprint 10-13
Add Vehicle	Vehicle registration and fleet management integration	Sprint 11-14
Add Driver	Driver registration with verification and compliance processes	Sprint 12-15
Add Collection Routes	Create and configure new collection routes with optimization	Sprint 13-16
Vehicle Management	Complete fleet administration with tracking and maintenance	Sprint 14-17
Admin Accounts	Administrative user account management with security controls	Sprint 15-18
Complete System Access	Full platform control with global administration oversight	Sprint 16-19

3.5. Platform Architecture Design with User Governance Framework Integration

3.5.1. Frontend Architecture with Role-Based User Experience Governance and Agile Development

The frontend architecture uses Next.js 15 with TypeScript to build modern React-based apps. It also uses comprehensive governance frameworks and Agile development practices to make sure that the user experience is good, that there is user role-based access control, that the app is accessible, which is that it is culturally appropriate, and that all stakeholders are happy, all while keeping security and performance standards high across all user categories (Vercel, 2024), and also the Frontend development uses Agile component-driven programming with React 18 and improves the user experience based on input from each sprint. Each sprint gives working software updates to certain groups of users, which lets stakeholders keep checking and make sure the program meets all the rules, and with the Test-Driven Development (TDD) is used for frontend components during the development process. This makes sure that the user interface works well for all four types of users.



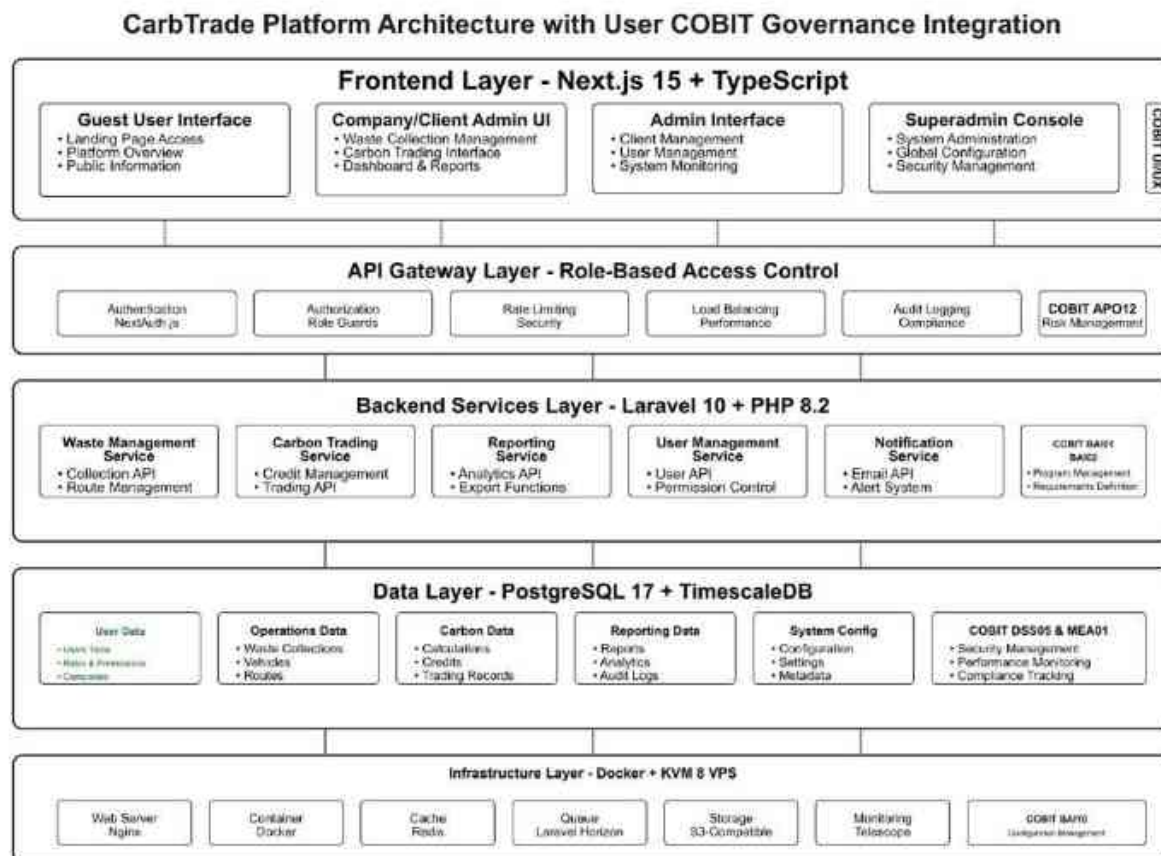


Figure 3.8. CarbTrade Platform Architecture with User COBIT Governance Integration

Figure 3.8 shows the full platform architecture with COBIT governance added to all levels of the architecture and all types of users. The architecture governance framework covers standards for frontend technology, protocols for backend development, procedures for managing databases, security requirements for integration, and best practices for managing infrastructures, and also these ensure that development is done in a systematic way with the right governance controls and accountability for all four user types.

Table 3.8. Frontend Architecture Components with User Governance Integration and Agile Development

Architecture Layer	Technology Stack	Key Benefits
Presentation Layer	Next.js 15 + TypeScript	Role-specific interfaces for all user types
Authentication Layer	NextAuth.js + RBAC	Secure hierarchical access control
Component Library	React 18 + ShadCN UI	Reusable components across user interfaces
State Management	Zustand + React Query	Role-specific data handling and performance
Routing System	Next.js App Router + Guards	Secure role-based navigation protection
Dashboard Framework	Custom Dashboard Components	User-specific dashboards for each role

3.5.2. Backend Architecture with User API Governance and Agile Development

The backend architecture is built on Laravel 10 and PHP 8.2, with strong governance frameworks and Agile development practices that make sure the API is reliable, user role-based security is in place, performance is improved, and integration is managed, It is also supports processing environmental data and meeting regulatory requirements for all user types (Laravel Foundation, 2024). Test-Driven Development (TDD) is used in Agile sprints to make sure that the backend API meets the needs of four different types of users. API endpoints are built in stages, and that with the governance controls which is built in at every stage of the development process, and this makes it possible to keep an eye on compliance and check for security issues at all times. The development method is based on Domain-Driven Design (DDD) and uses a microservices architecture to allow scalable user category functionality.

Table 3.9 Backend Architecture Components Supporting User Requirements with Agile Development

Architecture Layer	Technology Stack	Key Benefits
API Layer	Laravel 10 + PHP 8.2 + RESTful	Role-specific endpoints with secure access controls
Authentication System	Laravel Sanctum + RBAC	Multi-user authentication with audit trails
Authorization Layer	Laravel Policies + Guards	Fine-grained permissions for all user categories
Data Processing	Laravel Models + Role-Based Access	Secure data handling with privacy compliance
Queue Management	Laravel Horizon + Redis	Role-based processing with optimized performance
Audit System	Laravel Activity Log + Tracking	Comprehensive audit trails for regulatory compliance

3.5.3. Database Architecture with User Data Governance and Agile Development

The database architecture uses PostgreSQL 17 with TimescaleDB extensions. It is backed up by strong data governance frameworks and Agile development practices that guarantee data quality, user role-based security implementation, performance optimization, and compliance with regulations. It also supports processing of environmental time-series data and reporting requirements for all user categories. When developing a database, we use Agile database design concepts, which include iterative schema evolution, also an ongoing performance optimization, and with the data modeling based on governance. Every sprint involves database migrations, performance testing, and data governance validation to make sure that data management is scalable and secure for all types of users.

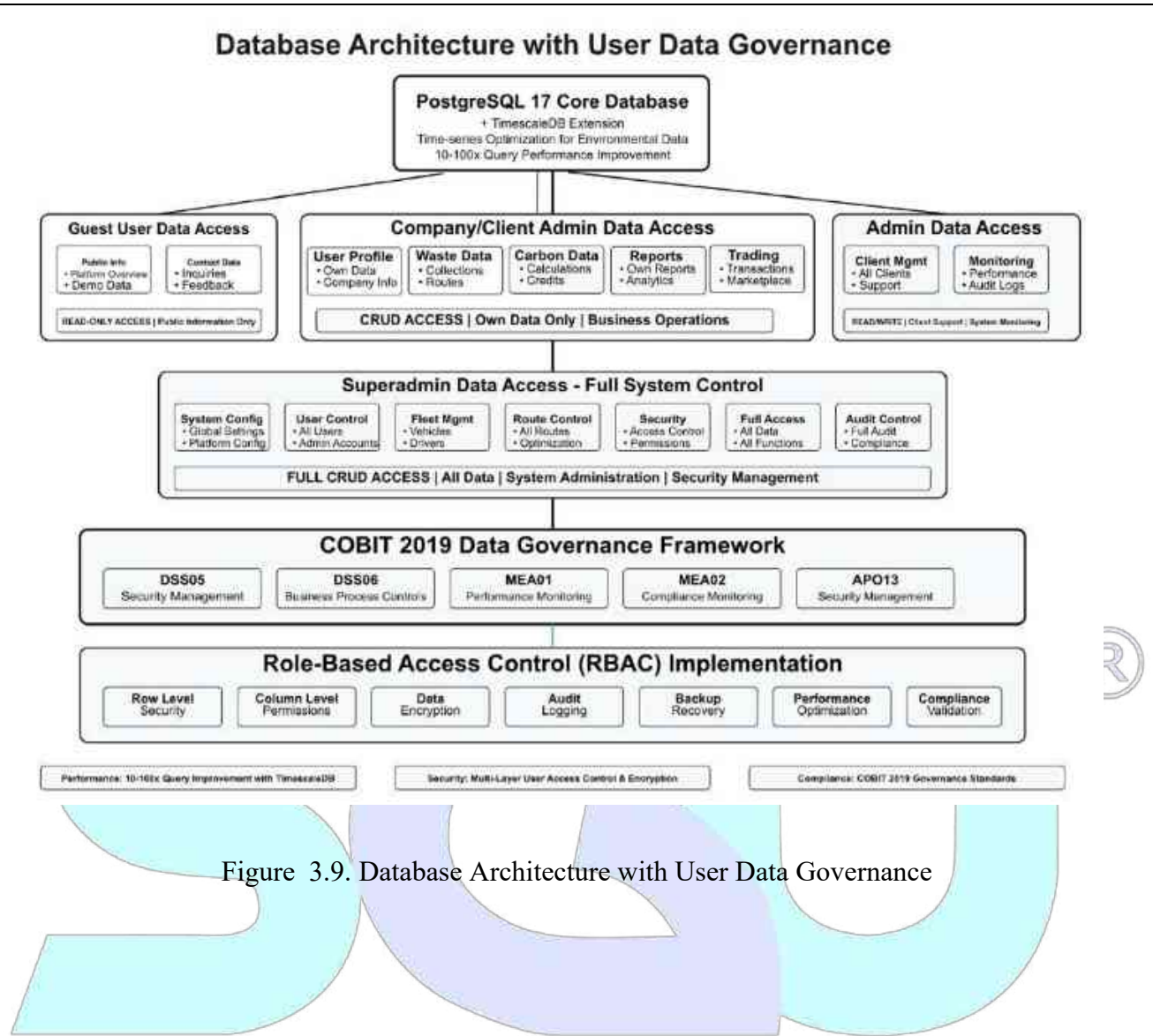


Figure 3.9. Database Architecture with User Data Governance

Table 3.10 Database Schema Design Supporting User Requirements with Agile Development

Database Entity	User Category Access	Key Benefits
Users Table	All categories with role-based restrictions	Secure user management across all user types
Roles and Permissions	Admin and Superadmin only	Granular access control and security management
Companies Table	Company/Client (own), Admin/Superadmin (all)	Secure client relationship management
Waste Collections	Company/Client (own), Admin (monitoring), Superadmin (full)	Accurate tracking with appropriate access controls
Carbon Calculations	Company/Client (own), Admin (support), Superadmin (full)	Reliable carbon data with role-based access
Carbon Credits	Company/Client (own), Admin (support), Superadmin (management)	Secure transactions with appropriate oversight
Reports Table	Company/Client (own), Admin (support), Superadmin (all)	Comprehensive reporting with access controls
Vehicles Table	Company/Client (assigned), Admin (monitoring), Superadmin (full)	Efficient fleet management with role-based access
Collection Routes	Company/Client (assigned), Admin (monitoring), Superadmin (full)	Optimized routing with appropriate access levels
Audit Logs	Admin (monitoring), Superadmin (full audit)	Complete audit trails with role-based access
System Configuration	Superadmin exclusive access	Secure system management with appropriate controls

3.5.4. Integration Architecture and User Ecosystem Connectivity with Agile Development

The integration architecture makes it possible for all external systems to connect to the carbon trading marketplace, which is necessary for its operation, with which it is also uses governance frameworks and Agile development practices to make sure that all integration interfaces are secure, reliable, and compliant for all user access levels (Joutsenlahti et al., 2021), and with the integration development which uses a Agile integration patterns, which include iterative API creation, ongoing testing, and

security implementation that is driven by governance. Each sprint includes testing for integration, security, and performance to make sure that all external systems and user groups can connect reliably.

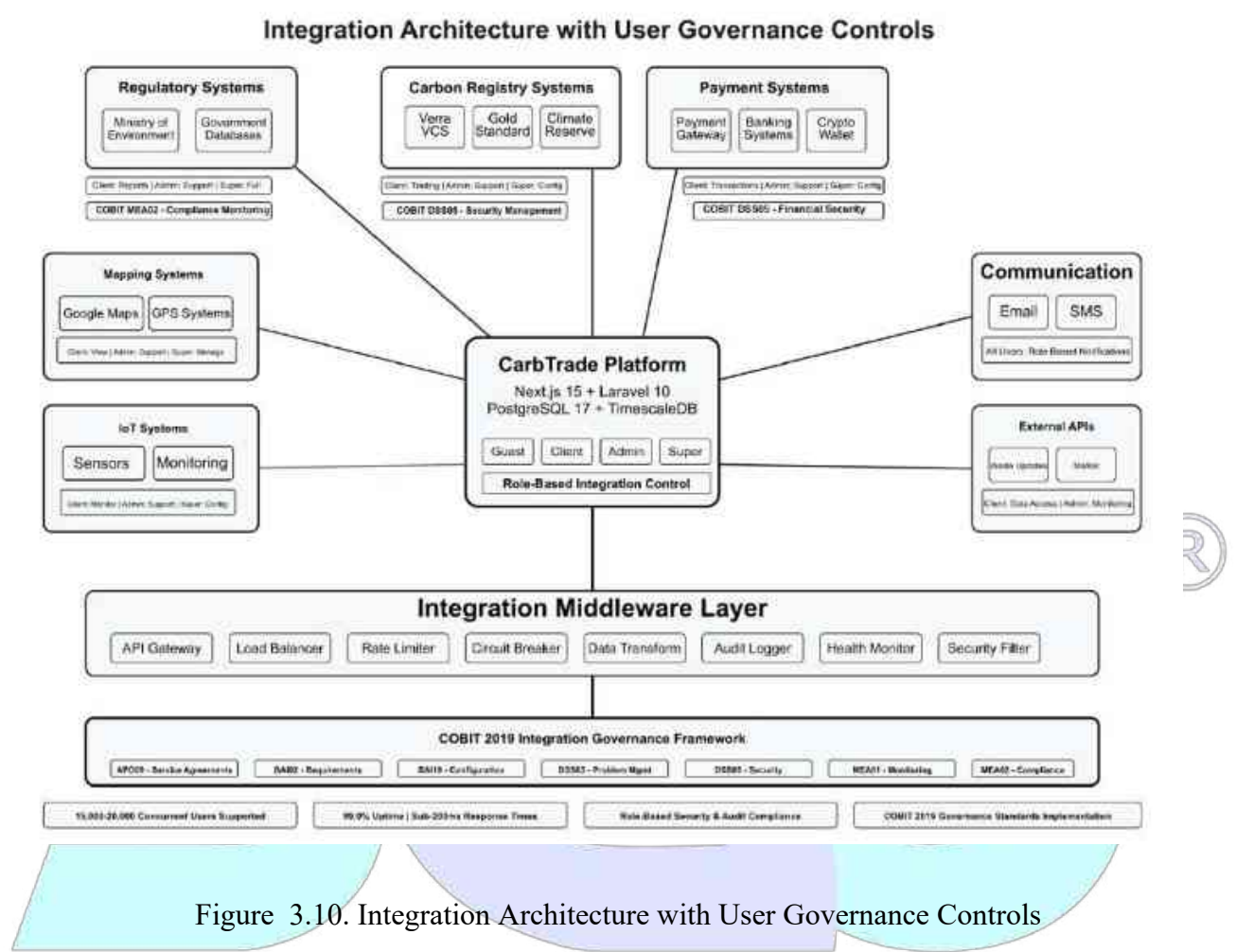


Figure 3.10. Integration Architecture with User Governance Controls

Table 3.11 Integration Architecture Supporting User Platform Requirements with Agile Development

Integration Type	User Category Access	Key Benefits
Regulatory Integration	Company/Client (reporting), Admin (support), Superadmin (full)	Secure regulatory compliance with appropriate access
Carbon Registry	Company/Client (transactions), Admin (support), Superadmin (config)	Secure carbon credit operations with role-based access
Payment Integration	Company/Client (transactions), Admin (support), Superadmin (config)	Secure financial transactions with appropriate oversight
Mapping Integration	Company/Client (viewing), Admin (support), Superadmin (management)	Optimized routing with role-based access controls
IoT Integration	Company/Client (monitoring), Admin (support), Superadmin (management)	Real-time monitoring with appropriate access levels
Communication Integration	All categories with role-appropriate access	Effective communication with role-based controls

3.6. Research Hypotheses and Theoretical Model Development

3.6.1. Hypothesis Development Framework with User Governance Integration

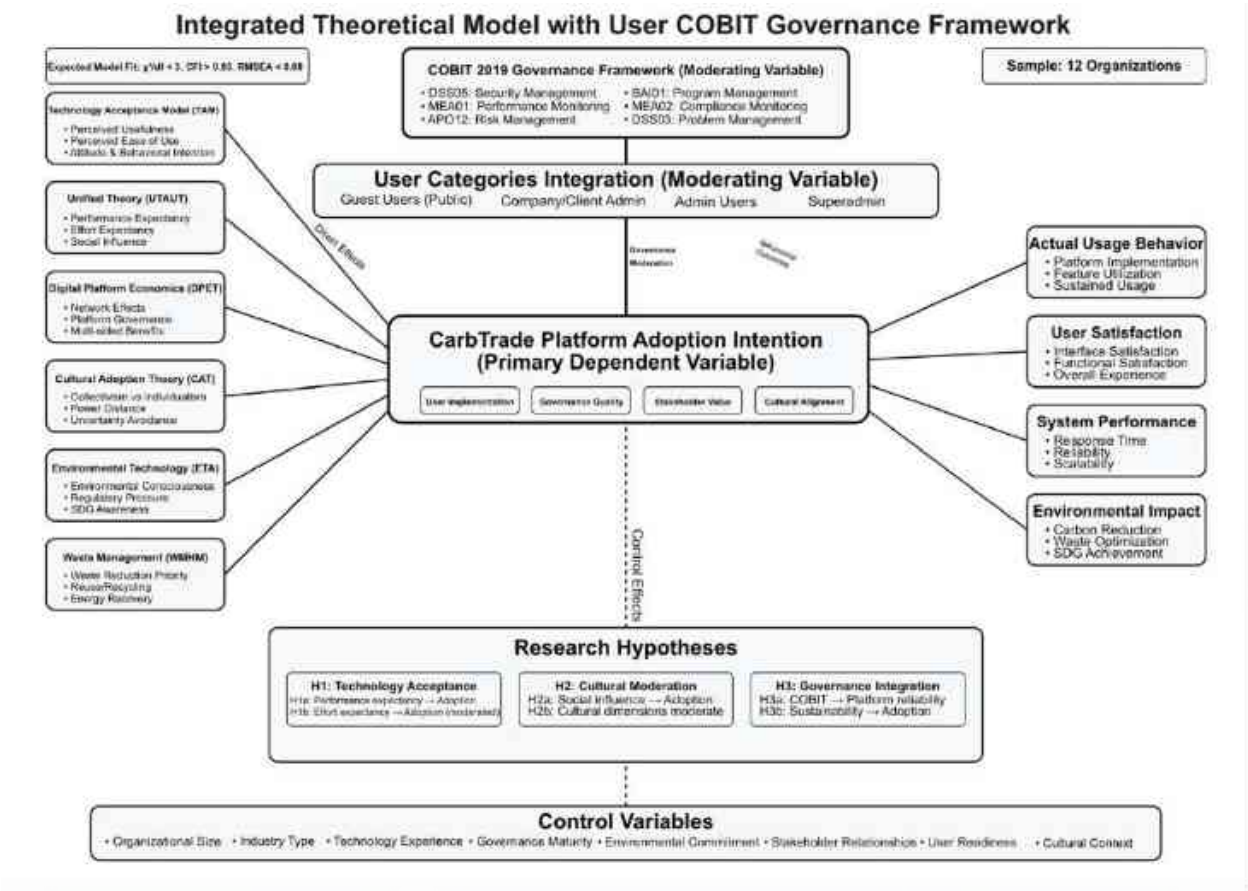


Figure 3.11. Integrated Theoretical Model with User COBIT Governance Framework

The process of developing a hypothesis involves systematic theoretical synthesis based on a thorough review of the literature and real-world findings from recent research on technology adoption in environmental and also developing country settings, and this is improved by integrating governance frameworks and as well as validating the hypothesis with four levels of users: Guest, Company/Client Admin, Admin, and Superadmin (Venkatesh, Thong and Xu, 2016). Figure 16 shows that three main hypotheses were created by carefully looking at how core technology acceptance constructs, platform-specific factors, Indonesian cultural moderators, governance quality indicators, and environmental context variables interact with each other at different user access levels.

H1: Technology Acceptance and User Platform Governance Quality Integration

H1a: Performance expectancy positively affects adoption intention across all four user types, with stronger effects for environmentally-conscious companies with mature governance.

H1b: Effort expectancy negatively influences adoption intention across all user groups, but this effect is moderated by user role complexity, organizational size, technical capabilities, and governance support availability.

H1c: COBIT governance framework quality positively moderates the relationship between performance expectancy and adoption intention across all user groups, with stronger effects for administrative users.

H2: Social Influence and Cultural Moderation Effects Enhanced by User Governance Credibility

H2a: Social influence positively affects adoption intention and usage across all user types, stronger in Indonesian collectivist contexts.

H2b: Indonesian cultural dimensions moderate effort expectancy-adoption relationships differently across user categories.

H2c: Facilitating conditions moderate intention-usage relationships across user categories with different requirements per user type.

H3: User Platform Governance and Sustainable Development Integration Effects

H3a: COBIT implementation improves platform reliability and trust across all users, stronger for Admin/Superadmin.

H3b: Sustainability alignment enhances adoption motivation across all user types, varying by role.

H3c: Governance quality mediates sustainability-adoption relationships, stronger for decision-making roles.

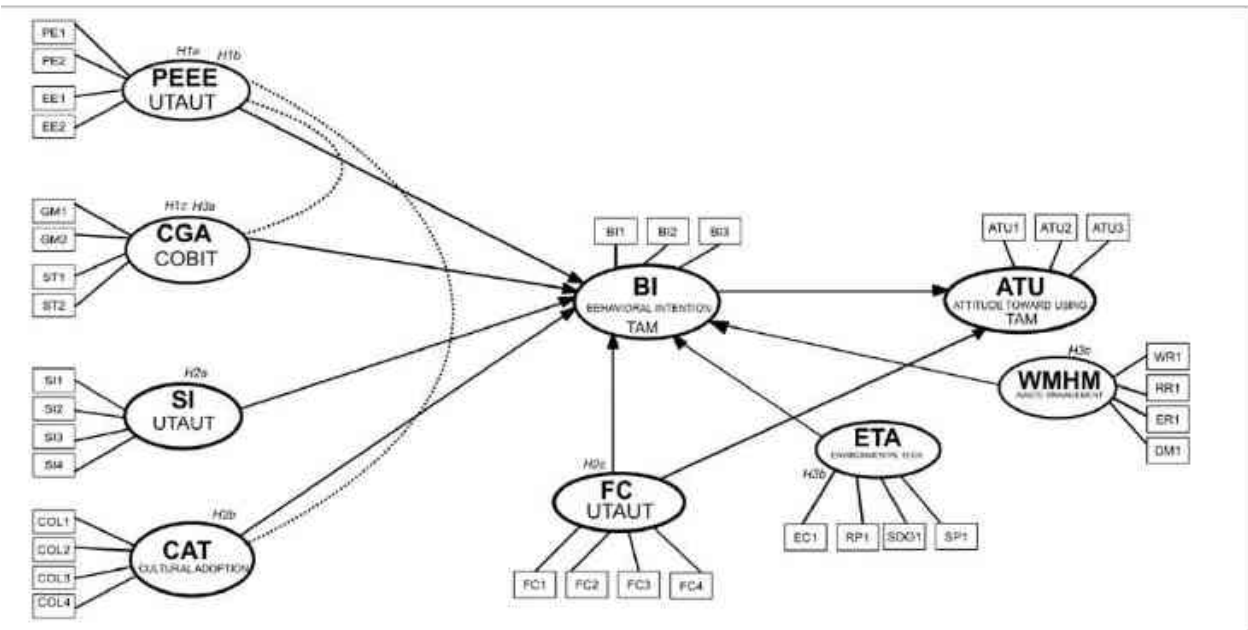


Figure 3.12. Research Model of the Study with Hypotheses



3.7. Population and Sampling Strategy

3.7.1. Target Population Definition with User Characteristics

The target population consists of the 15 current clients of Jangjo Technology Indonesia involved in waste management services and environmental sustainability initiatives, and these organizations possess established waste management operations, exhibit a commitment to sustainable practices, and display varying degrees of governance maturity across four user categories and organizational roles (Population Definition Research, 2024). This population offers pertinent context for comprehending the dynamics of CarbTrade adoption within established service relationships characterized by trust, operational experience, environmental commitment, and governance awareness across various user access levels and functional responsibilities.

Table 3.12 Target Population and control Variables

Demographic Category	Classification	Options
Industry Sector	Sector Type	Waste Management
		Manufacturing
		Commercial Districts
		Hospitality
		Education
		Retail/Services
		Other
Company Size	Employee Count	0-50
		50-200
		200-500
		500-1000
		>1000
Service Duration with Digital Platforms	Experience Level	<1 year
		1-2 years
		3-5 years
		>5 years
		Never
Formal Sustainability Policy	Implementation Status	Yes (implemented)
		Yes (developing)
		No (but planning)
		No plans
Digital Platform Experience	Technical Proficiency	Beginner
		Basic
		Intermediate
		Advanced
		Expert

3.7.2. User Sampling Method and Focus Group Discussion Design

This study uses a comprehensive mixed-sampling approach that includes census sampling for collecting quantitative data from all four user groups and purposive sampling for qualitative focus group discussions, and this gives the full picture of how the platform adoption works, taking into account governance issues and user validation (Tashakkori and Teddlie, 2010). The qualitative part includes Focus Group Discussions (FGD) with carefully chosen experts to get a deeper understanding of the problems with implementing governance, the problems with adopting technology across the four user groups, multi-user validation, and strategic suggestions for developing and deploying the platform to meet the needs of different users (Morgan, 2019).

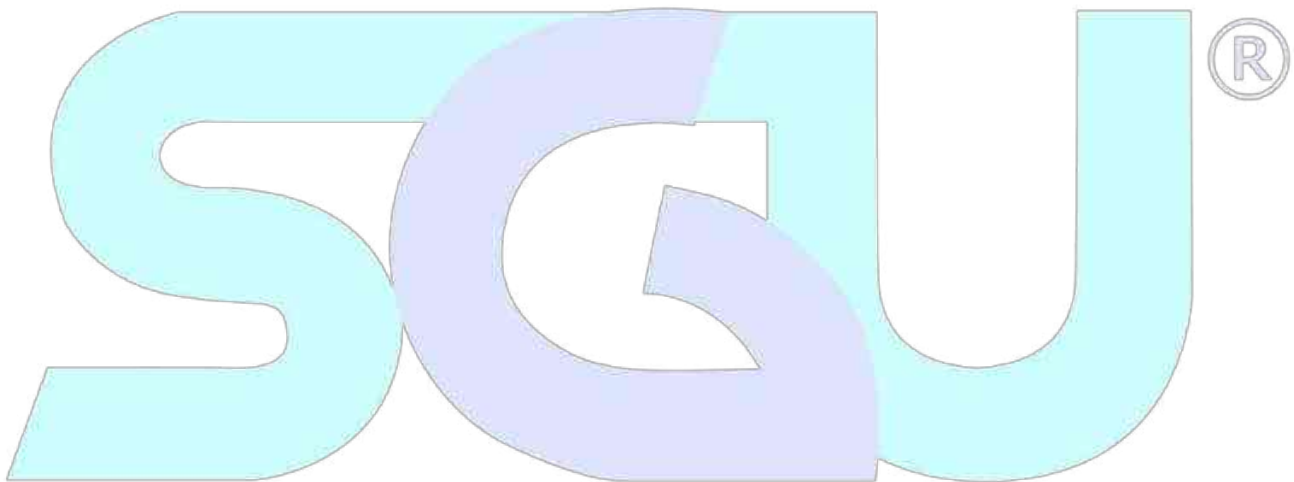


Table 3.13 Focus Group Discussion Expert Panel with User Perspective

Expert Position	Organization/Background	Expertise Areas	User Validation Role
Expert 1: Industry Operations	Jangjo Technology Indonesia COO	Waste management operations, client relationships, user technology implementation	Operational validation across Company/Client Admin and Admin user categories, Superadmin functionality assessment
Expert 2: Regulatory/Policy	Ministry of Environment and Forestry Representative	Environmental regulations, compliance requirements, multi-stakeholder policy frameworks	Regulatory validation for all four user categories, compliance requirement verification across access levels
Expert 3: Academic/Research	University Professor (Environmental Technology/IT Governance)	Technology adoption theory, governance frameworks, user system research	Theoretical validation across all user categories, research methodology verification for multi-tier analysis
Expert 4: Industry Association	Indonesian Waste Management Association Representative	Industry standards, best practices, user platform market dynamics	Industry perspective across all user categories, standards validation, market feasibility assessment

3.8. Questionnaire Development and Validation

3.8.1. Instrument Design with User Assessment Integration

The development of the questionnaire is based on established theoretical foundations and follows systematic psychometric principles. It also includes governance quality assessment and user familiarity evaluation across Guest, Company/Client Admin, Admin, and Super-Admin categories. Additionally,

it takes into account the specific needs of Indonesian organizations (Psychometric Research, 2024), and also the whole instrument has 110 construct questions spread out across seven theoretical constructs, as well as four tiers of user assessment, governance maturity evaluation, and demographic information. It has a total of 115 items and should take between 20 and 25 minutes to complete.

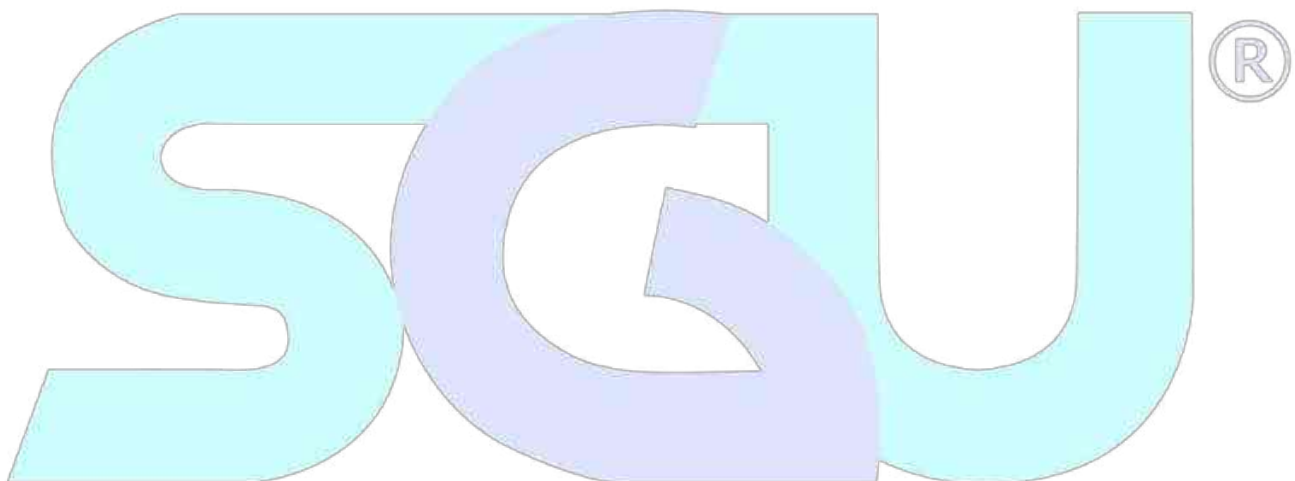


Table 3.14 Questionnaire Structure with User Assessment

Construct/Variable	Dimension	Observed Variable	Source
Technology Acceptance Model (TAM)	Perceived Usefulness	5 variables	Davis (1989), King & He (2006)
	Perceived Ease of Use	4 variables	
	Attitude Toward Using	4 variables	
	Behavioral Intention	3 variables	
		16 Variables	
Unified Theory of Acceptance and Use of Technology (UTAUT)	Performance Expectancy	4 variables	Venkatesh et al. (2003, 2022)
	Effort Expectancy	4 variables	
	Social Influence	4 variables	
	Facilitating Conditions	4 variables	
		16 Variables	
Digital Platform Economics Theory (DPET)	Network Effects	4 variables	Parker et al. (2016)
	Platform Governance	4 variables	
	Multi-sided Benefits	4 variables	
		12 Variables	
Cultural Adoption Theory (CAT)	Collectivism vs. Individualism	4 variables	Hofstede (2001)
	Power Distance	4 variables	
	Uncertainty Avoidance	4 variables	
	Long-term Orientation	4 variables	
		16 Variables	
Environmental Technology Adoption (ETA)	Environmental Consciousness	4 variables	Kumar et al. (2021)
	Regulatory Pressure	4 variables	
	SDG Awareness	4 variables	
	Stakeholder Pressure	4 variables	
		16 Variables	
Waste Management Hierarchy Model (WMHM)	Waste Reduction Priority	4 variables	Guerrero et al. (2013)
	Reuse/Recycling	4 variables	
	Energy Recovery	4 variables	

	Disposal Minimization	4 variables	
		16 Variables	
COBIT Governance Assessment (CGA)	Governance Maturity	4 variables	ISACA (2019)
	Risk Management	3 variables	
	Compliance Monitoring	3 variables	
	Stakeholder Trust	4 variables	
		14 Variables	
TOTAL		110 Variables	

3.8.2. Research Reliability

Scale development employs established measurement instruments adapted for environmental technology contexts with demonstrated reliability and validity across diverse cultural settings (Scale Development Research, 2024). Construct reliability assessment utilizes multiple criteria including Cronbach's alpha coefficients ($\alpha > 0.70$ for acceptable reliability, $\alpha > 0.80$ for good reliability), Composite Reliability ($CR > 0.70$), and Average Variance Extracted ($AVE > 0.50$) to ensure measurement quality and theoretical consistency (Reliability Assessment Standards, 2024).

Table 3.15 Construct Reliability

Variables	Cronbach's Alpha	Reference	This Study's Cronbach's Alpha
Technology Acceptance Model (TAM)	0.898	(King & He, 2006)	0.934
Unified Theory of Acceptance and Use of Technology (UTAUT)	0.77	(Venkatesh et al., 2003)	0.936
Digital Platform Economics Theory (DPET)	0.84	(Parker et al., 2016)	0.901
Cultural Adoption Theory (CAT)	0.83	(Hofstede, 2001)	0.927
Environmental Technology Adoption (ETA)	0.89	(Kumar et al., 2021)	0.927
Waste Management Hierarchy Model (WMHM)	0.81	(Guerrero et al., 2013)	0.939
COBIT Governance Assessment (CGA)	0.87	(ISACA, 2019)	0.938

3.9. Data Coding Framework and Variable Identification

3.9.1. Comprehensive Data Coding System with User Integration

Data coding uses a systematic alphanumeric identification system that makes statistical analysis easier while keeping respondents' information private, also data safe, and all variables tracked throughout the research process, and this is made possible by the user assessment integration and governance quality indicators (Data Coding Research, 2024), and each theoretical construct has standard prefixes, dimension indicators, and item numbers to make it easier to discover variables during analysis, and this is even better when user category assessment and role-specific governance evaluation are combined.

Table 3.16 Data Coding Framework with User Variable Identification

Variables	Dimensions	Observed Variables/ Indicators
TAM	Perceived Usefulness	PU1, PU2, PU3, PU4, PU5
	Perceived Ease of Use	PEOU1, PEOU2, PEOU3, PEOU4
	Attitude Toward Using	ATU1, ATU2, ATU3, ATU4
	Behavioral Intention	BI1, BI2, BI3
UTAUT	Performance Expectancy	PE1, PE2, PE3, PE4
	Effort Expectancy	EE1, EE2, EE3, EE4
	Social Influence	SI1, SI2, SI3, SI4
	Facilitating Conditions	FC1, FC2, FC3, FC4
DPET	Network Effects	NE1, NE2, NE3, NE4
	Platform Governance	PG1, PG2, PG3, PG4
	Multi-sided Benefits	MB1, MB2, MB3, MB4
CAT	Collectivism vs. Individualism	COL1, COL2, COL3, COL4
	Power Distance	PD1, PD2, PD3, PD4
	Uncertainty Avoidance	UA1, UA2, UA3, UA4
	Long-term Orientation	LTO1, LTO2, LTO3, LTO4
ETA	Environmental Consciousness	EC1, EC2, EC3, EC4
	Regulatory Pressure	RP1, RP2, RP3, RP4
	SDG Awareness	SDG1, SDG2, SDG3, SDG4
	Stakeholder Pressure	SP1, SP2, SP3, SP4
WMHM	Waste Reduction Priority	WR1, WR2, WR3, WR4
	Reuse/Recycling	RR1, RR2, RR3, RR4
	Energy Recovery	ER1, ER2, ER3, ER4
	Disposal Minimization	DM1, DM2, DM3, DM4
CGA	Governance Maturity	GM1, GM2, GM3, GM4
	Risk Management	RM1, RM2, RM3
	Compliance Monitoring	CM1, CM2, CM3
	Stakeholder Trust	ST1, ST2, ST3, ST4

3.9.2. Research Variables

3.9.2.1.Independent Variables with User Integration

The eight theoretical constructs that make up the integrated adoption framework are the independent variables, and also each one was measured using validated multi-item scales that were adapted for Indonesian cultural contexts and environmental technology applications, and these scales were also improved by a user familiarity assessment (Independent Variable Research, 2024).

Table 3.17. Independent Variable Classification with User Integration

Variable Category	Constructs	Reference
Digital Platform Economics Theory (DPET)	Network Effects, Platform Governance, Multi-sided Benefits	(Parker et al., 2016)
Cultural Adoption Theory (CAT)	Collectivism, Power Distance, Uncertainty Avoidance, Long-term Orientation	(Hofstede, 2001)
Environmental Technology Adoption (ETA)	Environmental Consciousness, Regulatory Pressure, SDG Awareness, Stakeholder Pressure	(Kumar et al., 2021)
Waste Management Hierarchy Model (WMHM)	Waste Reduction, Reuse/Recycling, Energy Recovery, Disposal Minimization	(Guerrero et al., 2013)
COBIT Governance Assessment (CGA)	Governance Maturity, Risk Management, Compliance Monitoring, Stakeholder Trust	(ISACA, 2019)

3.9.2.2. Dependent Variables with User Success Metrics

The main dependent variable is the intention to use the CarbTrade platform. This is improved by levels of the user implementation success indicators and an assessment of governance effectiveness that uses Behavioral Intention constructs from both the TAM and UTAUT frameworks. These have been adapted for environmental platform (Steininger *et al.*, 2014).

3.10. Data Collection Procedures and Implementation

3.10.1. Multi-Phase Data Collection Protocol with User Validation and Agile Integration

Data collecting is a very important step that has to be done in a systematic way to make sure that the data is accurate, ethical, and covers as many responses as possible within Jangjo Technology Indonesia's client ecosystem (Data collecting Research, 2024), as seen in Figure 3.11 shows that the multi-phase approach combines quantitative survey distribution with qualitative focus group discussions, user validation sessions, and governance maturity to assessment and to give a full picture of how technology adoption works, using Agile validation methods to make it even better.

Phase 1: User Validation and Platform Demonstration with Agile Testing (Week 1)

The first phase uses the Agile validation process and features user testing sessions that happen after each sprint cycle, and each validation iteration takes into account user input, governance assessment findings, and improvements to platform functionality, which this allows for ongoing improvements in both user experience and compliance effectiveness, and the platform demonstration sessions involve role-based scenarios for each user group, gathering input from stakeholders, and evaluating how well governance works.

User validation via platform demonstration sessions with representative users from each organizational group to make sure that the use case is correct, the capability is relevant, and the implementation is possible for all user categories (Shankarmani *et al.*, 2012). There are platform demonstration sessions that include:

- 1 **Guest User Demonstration:** how to go to the landing page, how to get an overview of the platform, and how to find information.
- 2 **Company/Client Admin Demo:** managing waste collection, using the carbon calculator, accessing reports, and selling carbon
- 3 **Admin Demonstration:** Managing users, helping clients, keeping an eye on the platform, and administrative controls
- 4 **Superadmin demonstration:** managing the system, configuring the whole system, managing security, and having full control over the platform

Validation sessions use structured feedback collection to assess user category clarity, functional requirement validation, workflow efficiency, and the implementation readiness, and also this

information is then used to improve the final questionnaire and research instrument (Shankarmani *et al.*, 2012).

Phase 2: Expert Focus Group Discussions with Agile Validation Integration (Week 2)

Focus Group Discussion implementation follows structured protocols designed to elicit comprehensive expert insights while maintaining engagement and ensuring data quality enhanced by user validation and governance framework discussion (Krueger and Casey, 2014). The 120-minute session employs the four-expert panel addressing user category validation, governance implementation challenges, technology adoption barriers, and strategic recommendations.

Table 3.18. Focus Group Discussion Protocol with User Validation and Agile Integration

Session Component	Duration	Focus Areas
Welcome and Overview	15 minutes	Research overview, user presentation, expert role clarification
User Validation Discussion	30 minutes	User category accuracy, implementation feasibility
Current Challenges and Adoption Barriers	25 minutes	Technology adoption barriers, governance challenges, cultural factors
Platform Architecture and Governance Assessment	25 minutes	Technical architecture evaluation, governance integration, scalability
Implementation Strategy and Recommendations	20 minutes	Deployment strategies, governance enhancement, cultural adaptation
Validation and Strategic Insights	5 minutes	Research validation, theoretical confirmation, practical insights

Phase 3: Quantitative Survey Distribution with Agile Data Collection (Weeks 3-6)

The primary data collection employs the Google Forms online survey platform, and with user assessment modules shared via a established communication channels, and these which included personalized invitations detailing user categories and benefits, and WhatsApp messaging for prompt notifications and follow-ups, including phone calls for relationship management and participation

encouragement, and in-person meetings as needed for senior management engagement and user category demonstration.

The survey distribution incorporates a pre-assessment of user readiness to guarantee comprehension of relevant terminology and understanding of organizational context, as well as the supplemented by platform demonstration materials and user category documentation that facilitate respondent understanding and precise evaluation (Ailawadi and Farris, 2017), and also Agile data collecting utilizes iterative response monitoring, weekly progress evaluations, prompt problem resolution, and ongoing stakeholder interaction to optimize response rates and data quality across all four user groups.

Phase 4: Governance Maturity and User Assessment with Agile Evaluation (Week 7)

The assessment of organizational governance maturity utilizes the COBIT 2019 maturity model, tailored for environmental platform contexts and augmented with a user evaluation of user implementation capabilities, and this approach evaluates governance readiness, implementation capabilities, user support effectiveness, and opportunities for improvement among participating organizations (ISACA, 2019b). The combined assessment evaluates governance maturity levels across various domains, including the effectiveness of governance structures for user category support, the implementation of risk management for platform security, the capabilities of compliance monitoring for regulatory requirements, the quality of stakeholder engagement for multi-user operations, and the readiness for the user implementation, which is encompassing technical capability, resource availability, and organizational support (De Haes, Van and Debreceeny, 2020). The Agile evaluation technique enables ongoing assessment refining, prompt feedback incorporation, and iterative enhancement suggestions, so assuring precise governance maturity evaluation and actionable implementation assistance.

3.10.2. Ethical Considerations and Stakeholder Protection with User Confidentiality

Ethical issues include getting informed consent that clearly explains the purpose of the research, user validation implications, governance assessment requirements, voluntary participation with the right to withdraw, protecting data privacy and anonymity, securely storing data with access controls, and the sharing results with the organizations that is took part when the study is over (Wilkinson *et al.*, 2016) User confidentiality protection includes protecting competitive information security, such as details about how a platform works that could give some user groups an edge over others; and also the protecting implementation strategy, which keeps an organization's competitive position; and protecting

stakeholder relationships, which keeps business relationships and partnership arrangements safe while also supporting research goals and knowledge sharing (Wilkinson *et al.*, 2016).

The agile ethical framework makes sure that ethical compliance is always being checked, problems are fixed right away, and stakeholder protection is improved throughout the data collecting process by doing frequent ethical reviews and putting in place improvements.

3.11. Data Analysis Framework and Statistical Procedures

3.11.1. Comprehensive Statistical Analysis Platform with User Integration and Agile Analytics

We use SPSS version 29 for descriptive statistics, reliability assessment, preliminary analysis, and data preparation. For more advanced Structural Equation Modeling (SEM) capabilities that are necessary for studying complex relationships between multiple theoretical constructs, we use SPSS AMOS version 29. These capabilities are enhanced by governance quality indicators and user implementation success factors (Statistical Software Research, 2024). Figure 3.12 demonstrates that the comprehensive analysis framework amalgamates quantitative statistical analysis with qualitative insights derived from focus group discussions, user validation outcomes, and governance maturity assessments. In which this integration facilitates a profound understanding of technology adoption dynamics, augmented by the efficacy of user category implementation and the integration of governance frameworks across all four user types, and also the agile analytics strategy lets you improve your analysis over time, validate your findings more often, and get new insights right away. This helps you do research faster and make your analysis more accurate and useful all the time.

Table 3.19 Statistical Analysis Framework with User Integration and Agile Analytics

Analysis Stage	Statistical Procedures	Software
Data Screening	Missing data analysis, outlier detection, normality testing	SPSS 26
Descriptive Analysis	Means, standard deviations, frequencies, distributions	SPSS 26
User Validation	Frequency analysis, cross-tabulation, correlation assessment	SPSS 26
Reliability Assessment	Cronbach's alpha, item-total correlations	SPSS 26
Correlation Analysis	Pearson/Spearman correlations, significance testing	SPSS 26
Comparative Analysis	One-way ANOVA, post-hoc testing	SPSS 26
Factor Analysis	Confirmatory Factor Analysis (CFA)	AMOS 26
Structural Modeling	Structural Equation Modeling (SEM)	AMOS 26
Qualitative Integration	Thematic analysis, content analysis	Manual

3.11.2. User Implementation Success Analysis with Agile Evaluation

User implementation success analysis looks at how user category familiarity, implementation readiness, and technology acceptance results are related while taking into account organizational characteristics, governance maturity, and also the cultural aspects (User Analysis Research, 2024), and the study looks at how the complexity of user categories affects the intention to adopt, how clear roles affect the success of implementation, and how well users grasp functional requirements affects their continued usage of the platform across all four user types.

The agile assessment technique lets you keep refining your analysis, get instant insights, and increase the accuracy and usefulness of your analysis over time. This helps you quickly iterate on your study and provide evidence-based recommendations.

3.12. Research Limitations

3.12.1. Methodological Limitations with User Constraints and Agile Limitations

Some of the methodological limitations are that the cross-sectional design makes it hard to figure out if there is a causal link between user understanding and adoption outcomes; self-reported data could be biased by social desirability, especially in user category familiarity self-assessments; as well as the limited geographic scope makes it hard to apply the findings to other Indonesian regions with different levels of technology readiness; and the focus on established client relationships makes it hard to apply

the findings to other market contexts where platform familiarity varies across the four user categories (Shankarmani *et al.*, 2012).

Other problems with user validation are that it relies on demonstration-based validation, which might not show how complicated actual implementation is across user categories; which there might be bias in user understanding assessment based on organizational position and technical knowledge; there isn't enough time to do a full evaluation of user implementation; and cultural differences in how functional concepts are understood might make the assessment less accurate in different Indonesian organizational contexts (Shankarmani *et al.*, 2012)

And finally other of the problems with agile methodology are that it can lead to scope creep in iterative development cycles, it can be hard to keep governance controls consistent across fast sprint iterations, it can be hard to balance agile flexibility with strict research methodology requirements, and it can be hard to manage stakeholder expectations when the platform is always changing during the research period (Shankarmani *et al.*, 2012).



Table 3.20. Research Limitations with User Considerations and Agile Constraints

Limitation Category	Specific Limitations	Mitigation Strategies
Design Limitations	Cross-sectional design, single time point	Theoretical justification, future longitudinal research
User Limitations	Demonstration-based validation, implementation complexity assumptions	Multiple validation methods, implementation pilot testing
Sample Limitations	Single region, established relationships	Broader sampling, user readiness distribution analysis
Measurement Limitations	Self-reported data, user familiarity self-assessment	Expert validation, user demonstration verification
Cultural Limitations	Indonesian-specific context, user cultural interpretation variation	Cultural validation, user adaptation guidance
Agile Limitations	Iterative development complexity, governance consistency challenges	Structured agile framework, governance checkpoint integration
Technology Limitations	Platform development constraints, demonstration environment differences	Multiple validation environments, comprehensive testing protocols
Temporal Limitations	Research timeframe constraints, sprint cycle limitations	Extended validation periods, phased assessment approach