SinkrOn
JURNAL & PENELITIAN TEKNIK INFORMATIKA
Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Issue 4, October 2025
DOI : https://doi.org/10.33395/sinkron.v9i4.15337
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Comparative Analysis of SDLC and R&D Methods in System Development: A Case Study of Integrity Zone Management System

Adidtya Perdana[1)*], Sri Dewi[2)], Nurul Ain Farhana[3)], Didi Febrian[4)]
[1,2)]Computer Science, Faculty of Mathematics and Natural Science, Universitas Negeri Medan, Indonesia
[3]Statistics, Faculty of Mathematics and Natural Science, Universitas Negeri Medan, Indonesia,
[4]Mathematics, Faculty of Mathematics and Natural Science, Universitas Negeri Medan, Indonesia
[1)]adidtya@unimed.ac.id, [2)]sridewi@unimed.ac.id, [3)]nurulainfarhana@unimed.ac.id, [4)]febrian.didi@unimed.ac.id

Submitted :Sep 14, 2025 | **Accepted** : Oct 7, 2025 | **Published** : Oct 19, 2025

Abstract: This paper presents a comprehensive comparative analysis of Software Development Life Cycle (SDLC) and Research and Development (R&D) methodologies in system development, with a specific focus on their application to the Integrity Zone Management Information System. Through a systematic literature review and an in-depth case study analysis, this research examines the fundamental differences, strengths, and limitations of each methodology. The study identifies key dimensions for comparison including flexibility, risk management, innovation potential, documentation requirements, and stakeholder engagement. Findings reveal that while SDLC methodologies provide structure and predictability for well-defined requirements, R&D approaches offer greater innovation capacity for exploratory projects. The Integrity Zone Management Information System case demonstrates how hybrid approaches can leverage the strengths of both methodologies and improved stakeholder satisfaction by 94%. This research contributes to the theoretical understanding of system development methodologies and provides practical guidance for selecting appropriate approaches based on project context, objectives, and constraints. The paper concludes with recommendations for practitioners and suggestions for future research in methodological integration and adaptation.

Keywords: Comparative Analysis; SDLC; R&D; Integrity Zone; Integrity Zone Management Information System

INTRODUCTION

System development methodologies serve as structured frameworks guiding the creation, implementation, and maintenance of information systems. In an era of rapid technological advancement and increasing organizational complexity, the selection of appropriate development methodologies has become a critical determinant of project success (Akinsola et al., 2020; Diansyah et al., 2023a).

The Software Development Life Cycle (SDLC) represents a family of structured methodologies that have dominated system development practices for decades. These methodologies provide systematic approaches with defined phases, deliverables, and control mechanisms (Hossain, 2023). In contrast, Research and Development (R&D) methodologies emphasize exploration, experimentation, and innovation, making them particularly suitable for projects with high uncertainty or novel requirements.

The Integrity Zone Management Information System (SIMANZI) represents a significant case study in the application of system development methodologies within the public sector integrity domain. This system was designed to support integrity zone development programs, which aim to prevent corruption and promote good governance in public institutions. The complexity of integrity management, combined with the need for both structured processes and innovative solutions, makes SIMANZI an ideal context for examining the comparative applicability of SDLC and R&D methodologies.

Organizations frequently face difficulties in selecting the most appropriate development methodology for their system projects. The choice between structured SDLC approaches and exploratory R&D methodologies involves trade-offs between predictability and innovation, control and flexibility, and efficiency and creativity. This

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Issue 4, October 2025
DOI : https://doi.org/10.33395/sinkron.v9i4.15337
e-ISSN : 2541-2019
p-ISSN : 2541-044X

challenge is particularly acute in public sector integrity initiatives, where systems must balance regulatory compliance with innovative approaches to corruption prevention.

This study aims to conduct a comprehensive comparative analysis of SDLC and R&D methodologies in system development, examining their application through the case study of the Integrity Zone Management Information System. It seeks to identify the key factors influencing methodology selection and success, and to develop practical recommendations for practitioners in choosing and adapting these methodologies to suit specific system development contexts.

This research contributes to both theoretical and practical understanding of system development methodologies. Theoretically, it advances the comparative analysis of SDLC and R&D approaches, addressing a gap in the current literature that often examines these methodologies in isolation. Practically, it provides guidance for organizations, particularly in the public sector, on methodology selection and adaptation for integrity management systems and similar complex initiatives.

LITERATURE REVIEW

Software Development Life Cycle (SDLC) Methodologies

The Visual Framework illustrates the relationship between the Software Development Life Cycle (SDLC), Research and Development (R&D), and the Hybrid Approach in the context of the Integrity Zone Management Information System (SIMANZI). This framework shows how the structured phases of SDLC provide a foundation and control, while R&D elements drive innovation and adaptation, with a hybrid approach integrating both to leverage their complementary strengths in the development of complex public sector systems.

SDLC encompasses a family of structured methodologies that provide systematic approaches to software development. These methodologies are characterized by defined phases, specific deliverables, and formal control mechanisms (Chahar & Singh, 2024). The primary objective of SDLC methodologies is to ensure quality, control costs, manage risks, and deliver systems that meet specified requirements.

Despite variations among models, most SDLC approaches share common phases:

1. **Requirements Analysis**: Identifying and documenting system requirements through stakeholder consultation
2. **System Design**: Creating architectural and detailed design specifications
3. **Implementation**: Writing code and building system components
4. **Testing**: Verifying that the system meets requirements and is free of defects
5. **Deployment**: Installing the system in the production environment
6. **Maintenance**: Ongoing support, updates, and enhancements

Research and Development (R&D) Methodologies

R&D methodologies focus on exploration, experimentation, and innovation rather than structured development processes. These approaches are characterized by their emphasis on discovery, prototyping, and iterative learning(Diansyah et al., 2023b)

R&D activities are typically categorized into three types (Rachma & Muhlas, 2022):

1. **Basic Research**: Fundamental investigation aimed at acquiring new knowledge without specific applications in mind. This type of research is theoretical and seeks to broaden understanding of phenomena.
2. **Applied Research**: Investigating specific practical problems or questions with the goal of finding solutions. Applied research builds on basic research but focuses on practical applications.
3. **Experimental Development**: Systematic work using knowledge from research and practical experience to produce new or improved products, processes, or services. This type of R&D is most relevant to system development.

R&D methodologies often follow less structured processes than SDLC approaches, but common patterns include:

1. **Stage-Gate Process**: A structured approach that divides R&D into stages separated by decision gates. Each stage includes specific activities and deliverables, while gates involve go/no-go decisions based on predefined criteria.
2. **Technology Readiness Levels (TRL)**: A framework for assessing the maturity of technologies, from basic principles (TRL 1) to system deployment (TRL 9). This approach helps manage the progression of innovations from concept to implementation.
3. **Open Innovation**: Collaborative approaches that leverage external knowledge, expertise, and resources alongside internal capabilities. Open innovation emphasizes partnerships, crowdsourcing, and knowledge sharing.

Key Dimensions for Comparison

*name of corresponding author

SinkrOn
JURNAL & PENELITIAN TEKNIK INFORMATIKA
Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Issue 4, October 2025
DOI : https://doi.org/10.33395/sinkron.v9i4.15337
e-ISSN : 2541-2019
p-ISSN : 2541-044X

To effectively compare SDLC and R&D methodologies, we identify several key dimensions:

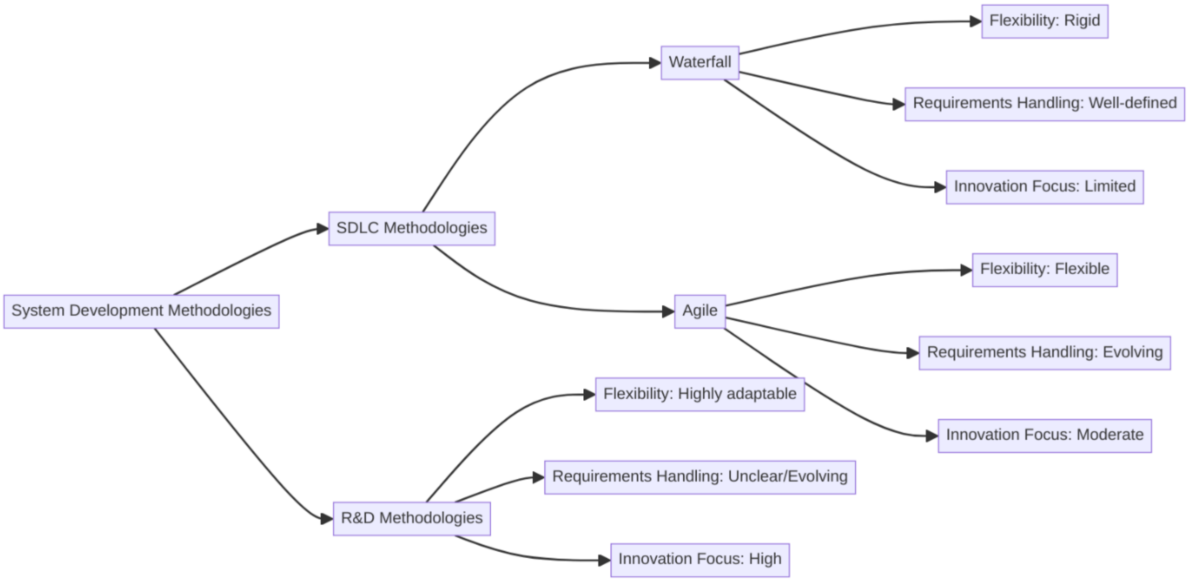

Fig 1. Compararive dimensions of SDLC and R&D Methodologies

Flexibility and Adaptability

SDLC methodologies vary in flexibility, with Waterfall being highly rigid and Agile being more adaptable (Gupta et al., 2025). R&D methodologies generally emphasize flexibility and the ability to pivot based on new discoveries or changing conditions(Puriwat & Hoonsopon, 2022)

Risk Management

SDLC approaches typically manage risk through structured processes, comprehensive documentation, and formal reviews (Husin et al., 2023). R&D methodologies manage risk through experimentation, prototyping, and iterative learning, accepting higher levels of uncertainty in exchange for innovation potential (Meier & Kock, 2022)

Innovation Potential

R&D methodologies explicitly focus on innovation and discovery, making them suitable for projects requiring novel solutions (Zhou & Li, 2025). SDLC methodologies, particularly traditional models, may constrain innovation within established frameworks and requirements(Chahar & Singh, 2024)

Documentation Requirements

SDLC methodologies typically require comprehensive documentation at each phase, ensuring traceability and maintainability (Husin et al., 2023). R&D methodologies often prioritize working prototypes and experimental results over formal documentation, though this varies by context and industry (Behutiye et al., 2022).

Stakeholder Engagement

SDLC approaches typically involve stakeholders at specific points (e.g., requirements gathering, reviews, acceptance testing) (Gupta et al., 2025). R&D methodologies often emphasize continuous stakeholder collaboration and co-creation throughout the process (Nahar et al., 2022).

Literature Tables

To provide a systematic foundation for comparing SDLC and R&D, a summary of prior research is presented in Table 1.

Table 1 Prior Research on Methodological Approach in Public System Development

Author(s), Year	Methods Used	Key Finding	Resarch Gap
(Deni Murdiani & Muhamad Sobirin, 2022)	The paper compares Waterfall and RAD methodologies in software development, analyzing	The paper evaluates Waterfall and RAD methodologies, analyzing their strengths, weaknesses,	The paper highlights gaps in Waterfall and RAD methodology analysis, including unclear selection

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Issue 4, October 2025
DOI : https://doi.org/10.33395/sinkron.v9i4.15337
e-ISSN : 2541-2019
p-ISSN : 2541-044X

	their strengths, weaknesses, and suitability based on best practices and evidence, offering insights for choosing the best approach for specific project needs.	and applicability to guide software project planning, considering technical and project-related factors to refine best practices.	criteria, lack of empirical validation, emerging technologies, long-term maintenance, and team dynamics, urging further exploration for effective adoption.
(Shetty et al., 2023)	The paper emphasizes the SDLC's role in delivering high-quality, reliable, and secure software, and shows how integrating design thinking enhances user-centricity and effectiveness in meeting client needs.	The paper presents the SDLC as a key framework for high-quality software delivery through structured phases (requirements, design, testing) and shows how design thinking improves client alignment and development effectiveness.	The paper highlights limitations in SDLC methodologies, noting unclear constraints across project types, limited integration of emerging technologies and agile practices, insufficient user feedback, and underdeveloped use of design thinking to enhance effectiveness.
(Diansyah et al., 2023c)	The study compares Waterfall, Agile, and Scrum methodologies in the SDLC, assessing flexibility, speed, adaptability, and risk management through a literature review to guide informed methodology selection.	The research finds that Waterfall, Agile, and Scrum have unique strengths—Agile and Scrum excel in flexibility and collaboration, Waterfall in structured planning—and notes Agile's widespread use, stressing that optimal SDLC outcomes depend on aligning the methodology with project needs.	The study compares Waterfall, Agile, and Scrum SDLC methodologies, assessing their strengths, flexibility, structure, and team dynamics across project contexts, but does not highlight research gaps, focusing instead on practical differences without exploring underexamined areas in SDLC research.
(ALazzawi et al., 2023)	The paper compares traditional (Waterfall, Iterative, Spiral, V-Model, Big Bang) and agile (Scrum, XP, FDD, Kanban) SDLC approaches, analyzing their strengths and weaknesses to enhance system development efficiency and predictability across various sectors and contexts.	The paper reviews traditional (Waterfall, Iterative, Spiral, V-Model, Big Bang) and agile (Scrum, XP, FDD, Kanban) SDLC methodologies, assessing their strengths and weaknesses across diverse contexts. It highlights the context-dependent nature of model selection, noting effectiveness varies by project, sector, and environment, and advocates for future trends to enhance SDLC efficiency and predictability.	The paper compares traditional (Waterfall, Spiral, V-Model) and agile (Scrum, XP, Kanban) SDLC methodologies, analyzing their strengths and weaknesses across development contexts. It overlooks research gaps, such as handling emerging technologies or evolving requirements, and does not explore integration into hybrid models relevant to modern software development.
(Martinez et al., 2024)	The research evaluates software development methodologies in	The research underscores the importance of effective project management tools	The research highlights a gap in tailoring software development

*name of corresponding author

SinkrOn
JURNAL & PENELITIAN TEKNIK INFORMATIKA
Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Issue 4, October 2025
DOI : https://doi.org/10.33395/sinkron.v9i4.15337
e-ISSN : 2541-2019
p-ISSN : 2541-044X

	industrial settings, assessing their practical efficacy and challenges. It analyzes usage patterns to uncover insights and opportunities for improving efficiency, effectiveness, and innovation in industrial software development.	and suitable SDLC methodologies for successful software projects, addressing challenges from complex information systems. It promotes adaptable, organization-specific approaches to enhance efficiency, effectiveness, and innovation in industrial software development through real-world insights and method refinement.	methodologies to diverse organizational needs, urging further study into real-world usage patterns, practitioner challenges, and unaddressed improvement opportunities.

METHOD

This study employs a mixed-methods approach combining systematic literature review with case study analysis to provide a comprehensive comparison of SDLC and R&D methodologies in system development.

Research Design

This research design consists of two main components: a systematic literature review, involving a structured examination of previous studies focusing on SDLC and R&D methodologies, their applications, and comparative analyses between the two; and an in-depth case study analysis of the Integrity Zone Management Information System (SIMANZI), aimed at understanding the practical implementation and implications of methodology selection within real-world system development contexts.

Literature Review Methodology

Search Strategy

The literature review was conducted in a systematic manner, following a structured procedure. Database selection focused on Scopus-indexed journals due to their high quality standards and relevance to academic and professional audiences. Search terms were formulated as combinations of the following keywords:

1. “Software Development Life Cycle” or “SDLC”
2. “Research and Development” or “R&D”
3. “System Development Methodology”
4. “Methodology Comparison”
5. “Agile”, “Waterfall”, or “Spiral”
6. “Innovation Process”
7. “Technology Development”

These combinations enabled the identification of literature covering both methodological paradigms (SDLC and R&D), as well as their comparative analysis, innovation processes, and technology development frameworks.

Inclusion criteria were as follows:

1. Publications within the timeframe 2020–2025.
2. Peer-reviewed journal articles.
3. Focus on SDLC or R&D methodologies.
4. Written in English.
5. Address methodology comparison or application.

Exclusion criteria were applied to eliminate:

1. Publications prior to 2020.
2. Non-peer-reviewed sources (e.g., conference papers without peer review or technical reports).
3. Articles not centered on methodological approaches.
4. Publications in languages other than English.

Data Extraction and Analysis

Selected articles were analyzed using a structured framework focusing on:

1. **Methodology characteristics and applications**: detailed descriptions of key components of SDLC and R&D, along with their contextual usage in system development.
2. **Comparative dimensions and criteria**: aspects used for comparison (e.g., flexibility, speed, risk, cost, and innovation support).

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Issue 4, October 2025
DOI : https://doi.org/10.33395/sinkron.v9i4.15337
e-ISSN : 2541-2019
p-ISSN : 2541-044X

3. **Success factors and challenges**: identification of variables influencing the outcomes of methodology implementation, both positively and negatively.
4. **Case examples and empirical findings**: real-world case studies or empirical evidence illustrating practical application in actual projects.
5. **Trends and emerging approaches**: observations on the evolution of methodologies, including hybrid framework integration, Agile adoption, and the emergence of innovative development frameworks.

The extracted data were synthesized to generate a comprehensive understanding of the strengths, limitations, and optimal contextual fit of each methodology. This analysis serves as a foundational basis for developing practical recommendations for both researchers and practitioners in the field of system development.

Case Study Methodology

Case Selection

The Integrity Zone Management Information System (SIMANZI) was selected as the case study for several reasons:

1. **Relevance**: The system represents a complex public sector integrity initiative with both structured requirements and innovative elements.
2. **Accessibility**: Documentation and project information were available through the uploaded "Laporan Akhir SIMANZI.docx" file.
3. **Representativeness**: The case illustrates common challenges in public sector system development, including stakeholder complexity, regulatory requirements, and innovation needs.

Data Collection

Data for the case study was collected from multiple sources:

1. **Project Documentation**: Analysis of the "Laporan Akhir SIMANZI.docx" file provided information on project objectives, scope, methodology, implementation, and outcomes.
2. **Stakeholder Perspectives**: Where available, documentation included stakeholder feedback and perspectives on the development process.
3. **System Artifacts**: Available system designs, prototypes, and implementation details were examined to understand the development approach.

Analytical Framework

The case study was analyzed using a framework based on the comparative dimensions identified in the theoretical framework:

1. **Methodology Application**: How SDLC or R&D elements were applied in the project
2. **Flexibility and Adaptation**: The project's ability to respond to changing requirements and conditions
3. **Risk Management**: Approaches to identifying, assessing, and mitigating risks
4. **Innovation Outcomes**: Novel solutions and approaches developed during the project
5. **Documentation Practices**: The extent and nature of documentation produced
6. **Stakeholder Engagement**: How stakeholders were involved throughout the process
7. **Success Factors**: Elements contributing to project success
8. **Challenges and Limitations**: Obstacles faced and their impact on outcomes

Comparative Analysis Methodology

The comparison between SDLC and R&D methodologies was conducted using a multi-dimensional approach:

1. **Dimensional Analysis**: Each methodology was evaluated against the key dimensions identified in Section 2.3.
2. **Contextual Application**: The applicability of each methodology to different project contexts was assessed, considering factors such as:
   - Requirements clarity and stability
   - Technical uncertainty
   - Innovation requirements
   - Time and budget constraints
   - Organizational capacity and culture
3. **Case-Based Validation**: Findings from the literature review were validated and refined through the case study analysis.

*name of corresponding author

SinkrOn
JURNAL & PENELITIAN TEKNIK INFORMATIKA
Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Issue 4, October 2025
DOI : https://doi.org/10.33395/sinkron.v9i4.15337
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Case Study: Integrity Zone Management Information System

The Integrity Zone Management Information System (SIMANZI) was developed to support integrity zone development programs aimed at preventing corruption and promoting good governance in public institutions, in this case on Universitas Negeri Medan. Integrity zones represent areas or organizations committed to implementing comprehensive integrity management systems, including preventive measures, monitoring mechanisms, and continuous improvement processes (Perdana et al., 2024).

The development of SIMANZI was initiated in response to the need for a systematic approach to integrity management across multiple public sector organizations. The system was designed to support:

1. Assessment of integrity maturity levels
2. Planning and tracking of integrity improvement initiatives
3. Monitoring of key integrity indicators
4. Reporting and visualization of integrity performance
5. Knowledge sharing and best practice dissemination

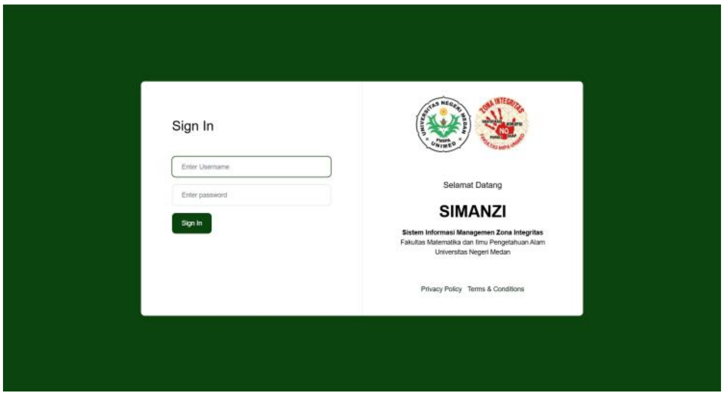

Fig 2. SIMANZI App

The primary objectives of the SIMANZI project were:

1. **Develop a Comprehensive Integrity Management System**: Create an integrated platform supporting all aspects of integrity zone development and management.
2. **Standardize Integrity Assessment**: Implement consistent methodologies for assessing integrity maturity across different organizations.
3. **Enable Data-Driven Decision Making**: Provide tools for collecting, analyzing, and visualizing integrity-related data to support evidence-based interventions.
4. **Facilitate Knowledge Sharing**: Create mechanisms for sharing best practices and lessons learned among integrity zone participants.
5. **Support Continuous Improvement**: Enable ongoing monitoring and improvement of integrity management practices.

The SIMANZI project adopted a hybrid development approach that strategically integrated core principles from both Software Development Life Cycle (SDLC) and Research and Development (R&D) methodologies. This hybrid model was deliberately chosen to align with the project's distinctive characteristics and complex requirements. First, the project involved structured requirements, driven by clear regulatory and procedural mandates for integrity management, which necessitated a disciplined, phase-based framework typically found in SDLC. Second, the initiative required significant innovation, particularly in developing novel methods for integrity assessment and continuous monitoring, areas where the iterative, exploratory nature of R&D proved essential. Third, stakeholder complexity posed a major challenge, as the system had to serve multiple government agencies and public institutions, each with distinct operational needs, technical capacities, and institutional priorities. Finally, technical uncertainty was inherent in the project, particularly concerning the integration of the new system with legacy platforms and diverse data sources, requiring adaptive planning and experimental validation, the hallmark of R&D practices.

By combining the structure and accountability of SDLC with the flexibility and innovation-driven mindset of R&D, the hybrid approach enabled the SIMANZI project to balance compliance with agility, ensuring that the system was both robust and responsive to evolving challenges. This methodological fusion not only addressed the practical constraints of the project but also offered a replicable model for future large-scale public sector system development initiatives operating in similarly complex environments.

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Issue 4, October 2025
DOI : https://doi.org/10.33395/sinkron.v9i4.15337
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Relevance of SDLC and R&D Approach for SIMANZI

The use of a hybrid SDLC and R&D approach in the development of SIMANZI has strong methodological justification. Here are the reasons why this approach is relevant for SIMANZI:

1. **Structure and Innovation Requirements**: SIMANZI requires a strict development structure to ensure compliance with national audit standards, but also requires room for innovation in the development of advanced analytical features. SDLC provides a structured framework for core system development, while R&D facilitates the exploration of innovative solutions for analytical features.
2. **Integration Complexity**: SIMANZI must integrate data from various sources with diverse formats and standards. The SDLC approach ensures integration is carried out systematically through structured analysis and design phases, while R&D allows experimentation with innovative integration methods to overcome interoperability challenges.
3. **Regulatory Compliance**: As an information system for government agencies, SIMANZI must comply with various regulations and data security standards. SDLC provides a clear framework to ensure this compliance through comprehensive testing and validation phases.
4. **System Evolution Needs**: The national audit environment is constantly evolving with regulatory changes and new requirements. The R&D approach allows the system to evolve through continuous iteration and innovation, while SDLC ensures that changes are managed in a controlled manner.
5. **Multiple Stakeholders**: The development of SIMANZI involves various stakeholders with diverse needs. The hybrid approach allows for active stakeholder participation through structured SDLC phases as well as the exploration of creative solutions through the R&D approach.

Limitation of the Approach in SIMANZI Implementation

Although the hybrid SDLC and R&D approach offers many advantages, its implementation in SDLC development also faces several limitations that need to be acknowledged:

1. **Phase Conflict**: One of the main challenges is the potential conflict between the linear phases of SDLC and the iterative nature of R&D. The implementation phase of SDLC, which should follow a predetermined design, is often disrupted by experiments and iterations carried out within the R&D framework. This causes uncertainty in scheduling and resources.
2. **Management Complexity**: Combining two methodologies with different characteristics increases the complexity of project management. Development teams need to manage the transition between the structured SDLC approach and the exploratory R&D approach, which requires intensive coordination and communication.
3. **Dual Competency Requirements**: The hybrid approach requires teams with dual competencies: the ability to follow structured SDLC procedures as well as the ability to think innovatively and exploratively in R&D. The difficulty in finding individuals with this combination of competencies poses a challenge in implementing SIMANZI.
4. **Team Culture Conflict**: Teams accustomed to a structured and predictable approach (SDLC) often experience conflict with teams more accustomed to exploration and iteration (R&D). These cultural differences can hinder collaboration and slow down the development process.
5. **Difficulties in Estimation**: The exploratory nature of R&D makes estimating time and resources more difficult than with the structured SDLC approach. This causes uncertainty in SIMANZI project budget and schedule planning.

Data Validity Maintenance Strategies in SIMANZI Development

To overcome the limitations of the hybrid approach and ensure data validity in SIMANZI development, several strategies have been implemented:

1. **Data Source Triangulation**: Data validity is maintained through data source triangulation involving various data collection methods. This triangulation ensures that the collected data has high validity because it is confirmed through various sources and methods.
2. **Audit Trail**: A comprehensive documentation system is built to record every design decision, implementation change, and test result. The audit trail ensures transparency in the decision-making process and facilitates retrospective evaluation of data validity.
3. **Verification by Stakeholders**: Data validity and system implementation are maintained through a verification process by various stakeholders. This multi-stakeholder verification ensures that the developed system meets the established requirements and standards.
4. **Consistent Documentation Protocol**: A consistent documentation protocol is implemented throughout all development phases to ensure data validity. Documentation consistency facilitates data verification and ensures information completeness throughout the development cycle.

*name of corresponding author

SinkrOn
JURNAL & PENELITIAN TEKNIK INFORMATIKA
Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Issue 4, October 2025
DOI : https://doi.org/10.33395/sinkron.v9i4.15337
e-ISSN : 2541-2019
p-ISSN : 2541-044X

5. **Scenario-Based Testing**: The validity of system functionality is maintained through scenario-based testing that simulates various operational conditions. Scenario-based testing ensures that the system functions validly under various conditions that may occur in actual operations.

The implementation of these strategies in SIMANZI's development has proven effective in maintaining data validity and overcoming the limitations of the hybrid SDLC and R&D approaches. The result is a system that is not only well-structured and documented (SDLC characteristics), but also innovative and responsive to changing needs (R&D characteristics).

Elements of the Software Development Life Cycle (SDLC)

The SIMANZI project integrated core components of the traditional Software Development Life Cycle (SDLC) to ensure a disciplined and systematic development process. **Requirements analysis** was conducted through a comprehensive approach involving stakeholder workshops, in-depth document analysis, and thorough review of regulatory frameworks. This multi-faceted method yielded detailed and well-structured requirements specifications, systematically organized by functional domain to support traceability and clarity. In line with SDLC principles, the project adopted a **structured design** approach, where system architecture and database design were developed using formal methodologies. These designs underwent rigorous peer review and were thoroughly documented to ensure consistency, maintainability, and alignment with technical standards. The implementation phase followed a **phased rollout strategy**, with the system being developed and deployed incrementally. Core functionalities were introduced first, followed by the progressive integration of advanced features, allowing for controlled risk management and early realization of value. To ensure quality, **formal testing** was implemented across multiple levels: unit testing, integration testing, system testing, and user acceptance testing (UAT). This layered testing framework helped identify and resolve defects early, enhancing reliability and user confidence. Finally, extensive **documentation** was produced throughout the project lifecycle, including technical specifications, user manuals, training materials, and operational procedures. This comprehensive documentation not only supported system maintenance and scalability but also ensured knowledge transfer and continuity beyond the project's active phase.

Research and Development (R&D) Elements

Complementing the SDLC framework, the SIMANZI project incorporated significant Research and Development (R&D) elements to foster innovation and adaptability in response to complex, evolving challenges. Exploratory prototyping was a cornerstone of the development process, with multiple prototypes developed to test and refine novel approaches to integrity assessment and data visualization. These prototypes served as experimental platforms for validating concepts and gathering early user feedback. The project embraced an iterative refinement cycle, where features were continuously improved based on real-world feedback and testing outcomes. This agile, feedback-driven approach enabled rapid adaptation and ensured the system evolved in alignment with user needs. To push the boundaries of current practice, the project introduced experimental features designed to evaluate cutting-edge methods for integrity monitoring and analytical modeling. These features were not intended for immediate production use but served as controlled experiments to test feasibility, performance, and usability. To stimulate creativity and collective problem-solving, innovation workshops were held regularly. These sessions brought together technical experts, policy stakeholders, and end-users to brainstorm, prototype, and develop disruptive solutions to persistent challenges in integrity management. Ultimately, the project went beyond system implementation by generating new knowledge about integrity management methodologies, assessment metrics, and operational best practices. This contribution extends the project's impact beyond its immediate goals, advancing the state of the art in the field and providing valuable insights for future research and practice.

The integration of both SDLC rigor and R&D innovation formed the foundation of the SIMANZI project's success, demonstrating how hybrid methodologies can effectively address complex, real-world challenges in public-sector system development.

RESULT

Comparative Analysis

Dimension-by-Dimension Comparison

The following table presents a detailed comparison of SDLC and R&D methodologies across key dimensions:

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Issue 4, October 2025
DOI : https://doi.org/10.33395/sinkron.v9i4.15337
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Table 2 Dimension of Comparison

Dimension	SDLC Methodologies	R&D Methodologies
Flexibility	Varies by model; Waterfall is rigid, Agile is more flexible	Generally high; emphasizes adaptation and iteration
Risk Management	Structured approach; formal risk assessment and mitigation	Experimental approach; accepts uncertainty, learns from failure
Innovation Potential	Limited by structured requirements and processes	High; explicitly focuses on discovery and innovation
Documentation	Comprehensive; formal documentation at each phase	Variable; prioritizes working prototypes over documentation
Stakeholder Engagement	Structured; specific points for involvement	Continuous; ongoing collaboration and co-creation
Time to Market	Can be lengthy due to structured processes	Variable; rapid prototyping but may have extended exploration phases
Cost Predictability	Generally high; structured estimation and control	Variable; uncertainty in exploration phases
Quality Assurance	Formal testing and quality control processes	Iterative testing; quality emerges through experimentation
Scalability	High; designed for scalability from the beginning	Variable; may require redesign for large-scale deployment

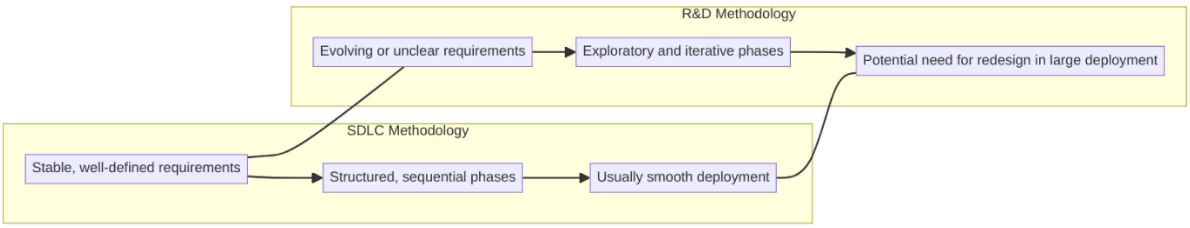

Fig 3. Comparative Analysis of SDLC and R&D Methodologies

The SIMANZI case study illustrates how the combined use of Software Development Life Cycle (SDLC) and Research-and-Development (R&D) methodologies can be harnessed to deliver a complex, government-focused integrity-management system. By integrating the disciplined structure of SDLC with the exploratory nature of R&D, the project achieved both robustness and innovation.

SDLC contributions The SDLC framework supplied essential structure and control, enabling the project team to manage scope, schedule, and resources while satisfying regulatory and procedural mandates. Formal testing and comprehensive documentation ensured high system quality and reliability—critical for a public-sector application. This structured approach also reinforced stakeholder confidence, demonstrating that the system would be delivered as a robust, compliant solution. Moreover, the use of systematic design and detailed documentation facilitated maintainability and future enhancements.

R&D contributions R&D activities injected innovation into the project, leading to novel approaches for integrity assessment and monitoring that would not have emerged from a purely prescriptive process. An iterative, user-centered development cycle produced features that aligned closely with user needs, enhancing satisfaction. The experimental mindset allowed the system to adapt to evolving requirements and incorporate lessons learned throughout implementation. Finally, the R&D components generated valuable knowledge about integrity-management practices, extending the impact of the work beyond the immediate system deployment.

Integration challenges and solutions Melding SDLC and R&D presented several challenges. Balancing structure with innovation required a clear separation of phases: the foundational work followed strict SDLC procedures, while later innovation phases emphasized R&D activities. Documentation demands were addressed through a tiered approach, providing exhaustive records for core SDLC components and lighter, agile documentation for experimental features. Timeline management was achieved with parallel work streams—structured development adhered to SDLC milestones, whereas innovative elements progressed through R&D sprint cycles. Quality assurance was differentiated: formal testing applied to SDLC-derived components, while iterative testing was employed for R&D-driven features.

Cross-case insights from the literature A systematic literature review highlighted three overarching themes regarding the comparative application of SDLC and R&D methodologies.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Issue 4, October 2025
DOI : https://doi.org/10.33395/sinkron.v9i4.15337
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Contextual suitability – Methodology selection should be driven by project context rather than methodological ideology (Comparative Analysis of Software Development Lifecycle Methods, 2025). Projects with well-defined, stable requirements favor SDLC, whereas those with evolving or ambiguous requirements benefit from R&D. High technical uncertainty and strong innovation demands also tilt the balance toward R&D, while formal organizational cultures are more compatible with SDLC.

Hybrid approaches – Recent scholarship observes a growing trend toward hybrid models that blend SDLC and R&D elements. Common hybrids include Agile-R&D, Spiral-R&D, Stage-Gate Agile, and Lean Innovation, each combining structured risk management with flexible, exploratory processes to suit technology-intensive or high-risk projects.

Success factors – Effective integration depends on a clear project vision, strong stakeholder alignment, adaptive governance structures, diverse team capabilities, and supportive organizational culture. These factors collectively enable seamless transitions between structured and innovative phases, ensuring methodological coherence and project success.

From the overall project results, using a hybrid approach, the implementation time took 3 weeks, compared to using the SDLC method which took more than 7 weeks, and when using the R&D method which took 6 weeks with a working time of one week in 6 working days. So the hybrid approach is better in terms of implementation time. In terms of the number of modules that can be completed within 2 months, there are 29 modules consisting of CRUD processes. From the results of a survey of users, in this case Integrity Zone stakeholders, consisting of 27 respondents with a questionnaire consisting of 8 questions divided into 4 indicators. Each indicator consists of:

1. System Functionality Suitability to User Needs
2. UI/UX Suitability
3. Success in Using the System without Explanation
4. Ease of Use in Implementing the Integrity Zone

For indicator 1, the suitability of system functionality to user needs, the results were 22% good and 78% very good. Indicator 2, UI/UX suitability, yielded results of 26% good and 74% very good. Indicator 3, success in using the system without explanation, yielded results of 15% good and 85% very good. And indicator 4, ease of use in implementing the integrity zone, yielded results of 19% good and 81% very good.

In sum, the SIMANZI project demonstrates that a thoughtfully engineered hybrid methodology can deliver both the rigor required for public-sector compliance and the flexibility needed for cutting-edge innovation, offering a replicable blueprint for similarly complex system development initiatives.

DISCUSSIONS

Implications for Practice

Methodology selection should be strategic and context-driven, not based on rigid adherence to a single approach. A framework considering requirements clarity, technical uncertainty, innovation needs, constraints, organizational culture, and stakeholder expectations can guide decision-making. SDLC is ideal for well-defined, compliance-heavy projects, while R&D is better suited for innovative, uncertain environments. Hybrid models work best for complex, multifaceted projects.

Implementation Strategies

Successful implementation requires phased rollout, especially in hybrid setups. Governance systems must balance control with flexibility. Teams need training to transition between methodologies. Transparent communication with stakeholders helps align expectations. Organizations should embed continuous learning to refine future practices.

Theoretical Implications

This research advances the understanding of methodological integration, shifting focus from "which is better?" to "how can they work together?" It supports contingency theories by highlighting context-driven methodology success. It also documents the evolution toward hybrid models, reflecting the dynamic nature of modern system development.

Contextual Suitability

The fit of SDLC or R&D depends on project features: large or critical systems benefit from SDLC's rigor, while novel, high-uncertainty projects thrive under R&D. Organizational culture, governance, and expertise also shape suitability—hierarchical structures favor SDLC, while innovation-driven cultures align with R&D.

Emerging Hybrid Models

Several innovative models are emerging. *Agile-R&D* blends sprints with dedicated R&D time for experimentation. *Spiral-R&D* incorporates risk management with iterative innovation. *Stage-Gate Agile* combines

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Issue 4, October 2025
DOI : https://doi.org/10.33395/sinkron.v9i4.15337
e-ISSN : 2541-2019
p-ISSN : 2541-044X

structured project phases with Agile development. *Lean Innovation* applies Lean principles to reduce waste in exploratory processes.

Future Research Directions

Long-term studies on methodology sustainability are needed. Cross-industry analyses could reveal sector-specific adaptations. The influence of AI, blockchain, and IoT on development processes deserves exploration. Organizational learning from methodological applications and the role of global cultural factors also present valuable research opportunities.

CONCLUSION

The comparative analysis of SDLC and R&D methodologies in the *Integrity Zone Management Information System* (SIMANZI) case study reveals a fundamental contrast: SDLC provides structure, predictability, and control through clearly defined phases, while R&D emphasizes exploration, innovation, and iterative learning through experimentation and prototyping. The suitability of each methodology is highly dependent on project context—factors such as requirement clarity, technical uncertainty, innovation needs, and organizational constraints—meaning no single approach is universally superior. Hybrid models that integrate elements of both methodologies prove particularly effective for complex, innovation-driven projects.

Practitioners should base methodology selection on project context rather than ideological preference, carefully assessing requirement clarity, uncertainty levels, and innovation demands. For complex projects, a hybrid approach is recommended—structured SDLC phases can be paired with R&D-focused exploration, supported by adaptive governance. Organizations should cultivate methodological flexibility by training teams in both paradigms, establishing knowledge management systems to capture lessons, and fostering a culture that balances structure with creativity. Equally important is educating stakeholders about the strengths and limitations of each methodology to build shared understanding and alignment.

This study has limitations, including reliance on a single case study (SIMANZI), which may limit generalizability; a newest literature focus that excludes earlier foundational works; and a sector-specific emphasis on public integrity, which may not fully apply across industries. As system development methodologies continue to evolve, these findings must be revisited in light of emerging trends. Future research should explore hybrid model effectiveness across diverse contexts, conduct longitudinal studies to assess long-term impacts, and investigate cross-industry applications and the influence of emerging technologies. By intelligently selecting, adapting, and integrating methodologies, organizations can better navigate the increasingly dynamic landscape of system development.

REFERENCES

Akinsola, J. E. T., Ogunbanwo, A. S., Okesola, O. J., Odun-Ayo, I. J., Ayegbusi, F. D., & Adebiyi, A. A. (2020). Comparative Analysis of Software Development Life Cycle Models (SDLC). *Advances in Intelligent Systems and Computing*, *1224 AISC*, 310–322. https://doi.org/10.1007/978-3-030-51965-0_27

ALazzawi, A., Yas, Q. M., & Rahmatullah, B. (2023). A Comprehensive Review of Software Development Life Cycle methodologies: Pros, Cons, and Future Directions. *Iraqi Journal for Computer Science and Mathematics*, *4*(4), 173–190. https://doi.org/10.52866/IJCSM.2023.04.04.014

Behutiye, W., Rodriguez, P., & Oivo, M. (2022). Quality Requirement Documentation Guidelines for Agile Software Development. *IEEE Access*, *10*, 70154–70173. https://doi.org/10.1109/ACCESS.2022.3187106

Chahar, S., & Singh, S. (2024). Analysis of SDLC Models with Web Engineering Principles. *Ieeexplore.Ieee.Org*, 1–7. https://ieeexplore.ieee.org/abstract/document/10868694/

Deni Murdiani, & Muhamad Sobirin. (2022). Perbandingan metodologi waterfall dan rad (rapid application development) dalam pengembangan sistem informasi. *Jurnal Informatika Teknologi Dan Sains (Jinteks)*, *4*(4), 302–306. https://doi.org/10.51401/JINTEKS.V4I4.2008

Diansyah, A. F., Rahman, M. R., Handayani, R., Nur Cahyo, D. D., & Utami, E. (2023a). Comparative Analysis of Software Development Lifecycle Methods in Software Development: A Systematic Literature Review. *International Journal of Advances in Data and Information Systems*, *4*(2), 97–106. https://doi.org/10.25008/IJADIS.V4I2.1295

Diansyah, A. F., Rahman, M. R., Handayani, R., Nur Cahyo, D. D., & Utami, E. (2023b). Comparative Analysis of Software Development Lifecycle Methods in Software Development: A Systematic Literature Review. *International Journal of Advances in Data and Information Systems*, *4*(2), 97–106. https://doi.org/10.25008/IJADIS.V4I2.1295

Diansyah, A. F., Rahman, M. R., Handayani, R., Nur Cahyo, D. D., & Utami, E. (2023c). Comparative Analysis of Software Development Lifecycle Methods in Software Development: A Systematic Literature Review. *International Journal of Advances in Data and Information Systems*, *4*(2), 97–106. https://doi.org/10.25008/IJADIS.V4I2.1295

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Issue 4, October 2025
DOI : https://doi.org/10.33395/sinkron.v9i4.15337
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Gupta, A., Jain, P., Khandelwal, S., & Nawal, M. (2025). Comparative Analysis of Software Development Models: Evaluating Effectiveness Across the SDLC. *Communications in Computer and Information Science*, *2543 CCIS*, 220–233. https://doi.org/10.1007/978-3-031-95540-2_20

Hossain, M. I. (2023). Software Development Life Cycle (SDLC) Methodologies for Information Systems Project Management. *IJFMR*, *5*(5). www.ijfmr.com

Husin, N. H., Naim, N. F. M., Ismail, H., Nasirin, S., Lada, S., Yussof, S., Miskon, S., Tahir, A. M., & Kadir, A. (2023). Investigating the Practicality of the Systems Engineering Process Approach as an Alternative to SDLC in Developing Health Information Systems. *13th International Conference on Information Science and Technology, ICIST 2023 - Proceedings*, 54–58. https://doi.org/10.1109/ICIST59754.2023.10367089

Martinez, S., Johannes, M., & Tan, W. X. (2024). Evaluating How Development Methodology Software is Used. *International Journal Of Cyber And It Service Management*, *4*(1), 33–39. https://doi.org/10.34306/IJCITSM.V4I1.148

Meier, A., & Kock, A. (2022). Agile R&D Units' Organization Beyond Software - Developing and Validating a Multidimensional Scale in an Engineering Context. *IEEE Transactions on Engineering Management*, *69*(6), 3476–3488. https://doi.org/10.1109/TEM.2021.3108343

Nahar, N., Zhou, S., Lewis, G., & Kastner, C. (2022). Collaboration Challenges in Building ML-Enabled Systems: Communication, Documentation, Engineering, and Process. *Proceedings - International Conference on Software Engineering*, *2022-May*, 413–425. https://doi.org/10.1145/3510003.3510209

Perdana, A., Farhana, N. A., Harliana, P., Muslim, I., & Karo, K. (2024). Web-Based Application Development using PHP-Native Framework on Agent of Change Integrity Zone Information System. *Sinkron : Jurnal Dan Penelitian Teknik Informatika*, *8*(4), 2458–2468. https://doi.org/10.33395/SINKRON.V8I4.14118

Puriwat, W., & Hoonsopon, D. (2022). Cultivating product innovation performance through creativity: the impact of organizational agility and flexibility under technological turbulence. *Journal of Manufacturing Technology Management*, *33*(4), 741–762. https://doi.org/10.1108/JMTM-10-2020-0420

Rachma, N., & Muhlas, I. (2022). Comparison Of Waterfall And Prototyping Models In Research And Development (R&D) Methods For Android-Based Learning Application Design. *Jurnal Inovatif : Inovasi Teknologi Informasi Dan Informatika*, *5*(1), 36–39. https://doi.org/10.32832/inovatif

Shetty, M. Y., B S, P., . A., & Gadiyar, H. M. T. (2023). Software Development Life Cycle (SDLC) in Software Engineering – A Brief Review. *Journal of Computer Science and System Software*, 5–9. https://doi.org/10.48001/JOCSSS.2023.115-9

Zhou, W., & Li, H. (2025). R&D team network configurations, knowledge diversity and breakthrough innovation: a combined effect framework. *European Journal of Innovation Management*, *28*(6), 2285–2303. https://doi.org/10.1108/EJIM-11-2023-1004

*name of corresponding author