Journal of Information Systems Engineering and Management
2025, 10(59s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Behavior-Driven Development (BDD) in Ruby on Rails for Healthcare QA Automation

Goutam Reddy Singireddy
Independent Researcher

ARTICLE INFO

Received: 12 July 2025
Revised: 15 Aug 2025
Accepted: 24 Aug 2025

ABSTRACT

The following technical article will review in detail how Behavior-Driven Development (BDD) techniques can be applied to Ruby on Rails to automate healthcare quality assurance. It will focus on the role that BDD collaborative methods can play in ensuring that there exists successful communication between technical teams and clinical stakeholders by promoting human-readable specifications. It researches the complementary uses of Cucumber to test workflow-level acceptance testing and RSpec to test with greater precision unit and integration testing in a healthcare setting. The article examines how these models overcome the peculiarities of healthcare software validation, such as patient safety validation, regulatory conformance, and audit trials confirmation. To inform practical insights on how best to implement BDD in clinical practice, the article relies on the wealth of academic literature to offer insights into practice implementation techniques, problems, and the best practices in the application of BDD in health care. Specific focus is on embodiment of complex clinical use cases, test coverage of safety-relevant tasks, verification against interoperability with external systems, handling of realistic test data and staying within privacy requirements. The results illustrate the high efficacy of the BDD methodologies to boost quality assurance procedures regarding healthcare applications in the eventual achievement of enhanced patient safety and the regulatory conformance.

Keywords: Behavior-Driven Development, Healthcare QA Automation, Cucumber, RSpec, Patient Safety Testing

1. Introduction

The use of healthcare software systems poses unique problems to those in quality assurance. These high-stakes applications must be subject to painstaking testing practices in order to safeguard patient care, comply with the strict regulations and ensure reliability in application operation. Behavior- Driven Development has emerged as a strong technique overcoming these problems in Ruby on Rails projects. This technical article digs into BDD implementation for healthcare QA automation, spotlighting how Cucumber and RSpec frameworks boost test coverage, confirm patient safety scenarios, and guarantee solid audit-trail validations across Electronic Health Record systems.

The healthcare sector's growing dependence on digital records adds fresh layers of complexity to software quality assurance. Research points out in analysis of health IT implementation hurdles that intricate clinical workflows paired with essential data processing needs to forge situations where software flaws directly harm patient outcomes [1]. Examinations of varied healthcare deployments show conventional testing methods repeatedly struggle with subtle clinical workflow requirements, especially those involving multi-disciplinary care teams and advanced decision support tools. Proper healthcare software validation must go beyond simple function checking to thoroughly assess clinical safety requirements, data exchange standards, and regulatory obligations.

Research expands this viewpoint, exploring specific advantages behavior-driven approaches deliver in regulated software environments [2]. Careful study across numerous software domains highlights particular healthcare application trials, where specifications typically spring from diverse stakeholders holding different system functionality expectations. Behavior-driven methods deliver major benefits through plain-language specifications understandable to clinical experts without technical knowledge.

Journal of Information Systems Engineering and Management
2025, 10(59s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

This joint approach effectively shrinks communication divides between development squads and healthcare practitioners, yielding broader testing coverage for vital clinical scenarios.
BDD deployment within Rails frameworks offers promising healthcare quality assurance possibilities. Cucumber's expressive, domain-specific language abilities enable test specifications mirroring clinical workflows, while RSpec provides exact control for complex calculations and data transformations crucial to healthcare applications. Research describes how natural language specification frameworks enable direct clinical stakeholder participation in validation activities, boosting test scenario precision and enhancing system reliability confidence [1]. This technique directly tackles a fundamental challenge: making sure software testing properly captures complex and sometimes subtle healthcare setting requirements.
The paper then proceeds to discuss concrete approaches to utilizing BDD in healthcare applications with a particular focus on strategies to map out complex clinical workflows, validate patient safety scenarios, comply with regulations and maintain comprehensive audit-trails. Combining the power of Cucumber acceptance testing with the unit and integration testing power of RSpec allows the healthcare organisations to design effective comprehensive testing strategies that address the domain-specific testing issues but retain the agility and efficiency advantages inherent in the modern software development approaches.

2. The BDD Approach in Healthcare Software Testing

Behavior-Driven Development is an extension of Test-Driven Development with the emphasis on collaboration between the developers and QA specialists as well as other non-technical participants. In the various contexts of healthcare this type of collaborative effort can be of particular value as healthcare subject matter experts can be central participants in testing activities in that they can directly contribute to testing activities via test specifications which are easy to read.BDD implementation within Ruby on Rails healthcare applications typically follows a structured path:

- Defining features and scenarios using plain language with Gherkin syntax
- Implementing step definitions linking scenarios to application code
- Running automated tests that validate expected behavior
- Creating comprehensive reports documenting compliance and test coverage

This structured testing approach marks a significant advancement in healthcare software quality assurance methods. Research studying software testing strategies across modern healthcare environments, stress that traditional testing approaches regularly fail to properly address the multifaceted, multi-stakeholder nature of healthcare applications [3]. Studies examining various quality assurance tactics highlight healthcare software presenting unique difficulties due to connected workflows, strict regulatory demands, and direct patient safety consequences. Effective healthcare testing methods must handle both technical validation and clinical verification—ensuring software works correctly from technical standpoints while accurately implementing intended clinical workflows and safety protections.BDD implementation across healthcare settings has demonstrated promising results tackling these challenges. Research documents in a thorough case study how BDD adoption within a healthcare technology firm significantly enhanced collaboration between technical and clinical teams [4]. The embrace of BDD practices transformed testing activities by establishing common language between developers, QA specialists, and healthcare professionals. This cooperative approach allowed clinical experts to directly review and confirm test scenarios before implementation, ensuring tests accurately reflected actual clinical workflows. Early validation processes caught numerous potential problems that might have slipped through using traditional testing approaches, particularly within complex areas like medication management and clinical decision support.BDD benefits in healthcare applications stretch beyond improved test coverage to include better documentation, traceability, and evidence generation for regulatory compliance. Research emphasizes healthcare software testing must produce comprehensive evidence of regulatory requirement compliance, including documentation of test coverage for critical patient safety scenarios [3]. The organized nature

Journal of Information Systems Engineering and Management
2025, 10(59s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

of BDD specifications creates natural connections between requirements and test cases, simplifying documentation processes for regulatory submissions. Similarly, research shows how BDD implementations strengthened organizational ability to respond to regulatory audits by delivering clear, readable documentation of test coverage for critical clinical features [4]. This documentation improvement represented major benefits beyond direct software quality enhancements, cutting administrative burdens tied to regulatory compliance activities.

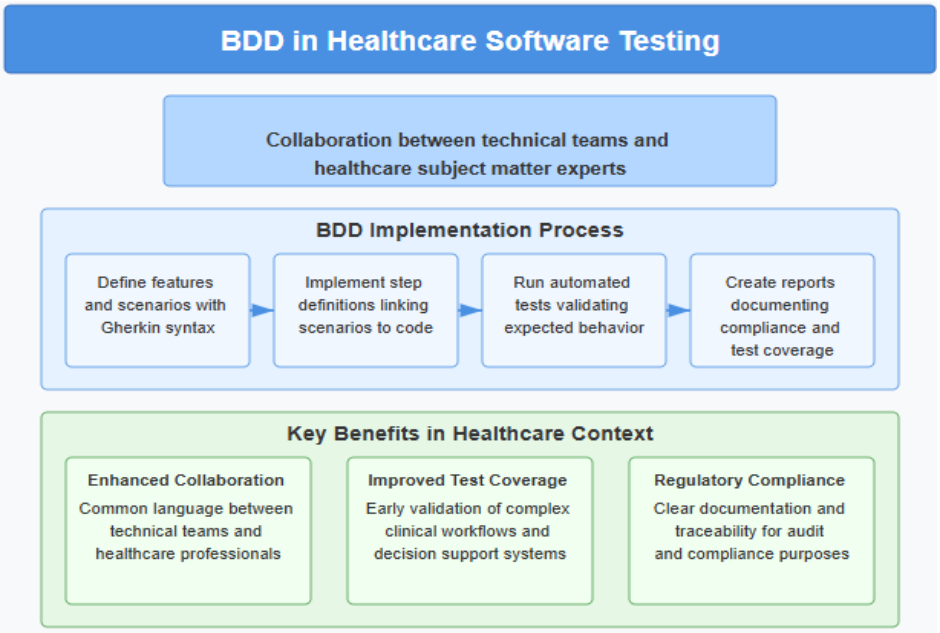

Fig 1: BDD Approach in Healthcare Software Testing [3, 4]

3. Cucumber for Healthcare Workflow Testing

Cucumber stands as an ideal framework for testing complex healthcare workflows through its ability to express scenarios using plain language that clicks with clinical stakeholders. The Gherkin syntax helps healthcare professionals verify test scenarios properly mirror clinical workflows without needing technical expertise.For example, medication administration workflows get described in straightforward language specifying expected behavior when nurses prepare to give medications to patients. These specifications cover patient identification verification, medication dosage confirmation, and proper documentation—all vital safety measures across healthcare settings.Applying Cucumber to healthcare testing environments marks a significant step forward in closing communication gaps between technical and clinical teams. Research through systematic review of technological interventions improving healthcare service delivery, stress the crucial importance of effective communication between technical developers and healthcare professionals during design and implementation of healthcare information systems [5]. Studies document numerous instances where mismatches between technical implementations and clinical workflows caused reduced system adoption, workflow bottlenecks, and potential patient safety hazards. Successful healthcare technology implementations demand methods enabling continuous validation by clinical stakeholders throughout development cycles. While not explicitly addressing BDD methodologies, these findings strongly back approaches like Cucumber that enable direct validation of software behavior by clinical subject matter experts without requiring technical know-how.Gherkin's structured specifications deliver additional healthcare testing advantages beyond improved stakeholder engagement. Studies examining electronic health record implementation challenges highlight clinical workflow complexity and difficulties capturing these workflows in traditional software requirements documents [6]. Examination of EHR implementation challenges across multiple healthcare organizations emphasizes how dynamic and

Journal of Information Systems Engineering and Management
2025, 10(59s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

context-dependent healthcare processes create significant software testing obstacles. Traditional testing methods often struggle to properly model complex, branching decision paths characterizing many clinical workflows. Though not directly addressing BDD frameworks, this research underlines needs for testing approaches expressing complex clinical scenarios in formats both machine-executable and human-readable—exactly what Cucumber provides through Gherkin syntax.Cucumber's capability to express complex clinical scenarios using natural language proves particularly valuable for testing healthcare workflows involving multiple stakeholders and systems. While research doesn't specifically address Cucumber or BDD methodologies, findings regarding multidisciplinary collaboration importance in healthcare software development strongly align with collaborative approaches Cucumber facilitates [5]. Emphasis on continuous validation by diverse stakeholders throughout development cycles supports tools making test scenarios accessible to non-technical clinical experts. Similarly, research on EHR implementation challenges highlights critical importance of effectively modeling complex clinical workflows typically involving multiple healthcare roles, decision points, and system interactions [6]. These findings indirectly support testing frameworks like Cucumber expressing complex workflows in formats accessible to all stakeholders while remaining precise enough to support automated testing.

Benefit Category	Cucumber	RSpec	Combined BDD Approach
Clinical Workflow Validation	High	Low	High
Technical Component Testing	Low	High	High
Patient Safety Verification	Medium	Medium	High
Regulatory Compliance	Medium	Medium	High
Stakeholder Collaboration	High	Low	High
Test Coverage	Medium	Medium	High
Documentation Quality	High	Medium	High
Error Detection Effectiveness	Medium	High	High

Table 1: Effectiveness of BDD Components in Healthcare QA Testing [5, 6]

4. RSpec for Unit and Integration Testing

Cucumber is great at acceptance-level testing whereas RSpec provides greater control to unit and integration tests. In all forms of healthcare use, RSpec is particularly useful in testing complex clinical calculations and algorithms, data transformation testing against interoperability standards such as HL7 and FHIR, required effective use of security controls, and verification of database integrity and audit trails.

These tests are more technical and ensure that lower level application elements in the healthcare work as intended especially in situations that involve critical procedures such as calculation of the medication dose which require certain precision to be ensured so that patient safety is not compromised.

The precision and flexibility RSpec offers make it an essential tool for validating critical technical components of healthcare applications. Research through comprehensive analysis of clinical decision support system malfunctions, document significant patient safety implications from software defects in healthcare applications [7]. Their investigation of 68 malfunction incidents affecting clinical decision support systems revealed computational inaccuracies represented a substantial portion of safety-critical defects. The study identified several computational error categories, including incorrect clinical algorithm implementation, failure to properly handle exceptional cases, and errors in unit conversion calculations. This analysis emphasizes computational defects often resist detection through manual testing or high-level acceptance tests alone, highlighting needs for rigorous unit testing of clinical algorithms. The authors conclude comprehensive testing strategies for healthcare applications must include both functional testing of clinical workflows and detailed verification of computational

Journal of Information Systems Engineering and Management
2025, 10(59s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

accuracy, particularly for high-risk functions such as medication dosage calculations and clinical alerting systems.

The importance of thorough testing for healthcare interoperability gains further emphasis through research examining health data exchange challenges [8]. Their analysis of interoperability in healthcare information systems stresses healthcare data fragmentation across different systems and standards represents a significant barrier to effective care coordination and clinical research. Two major problems observed by the study include the fact that successful interoperability does not rely only on structural compatibility between systems, but also on semantic consistency in the interpretation of clinical concepts. Successful interoperability implementation validation needs to demonstrate not only syntactic conformance to an interoperability standard such as HL7 and FHIR but also semantic fidelity at the transformation of clinical data across data formats and terminologies. These findings spotlight needs for testing tools capable of validating complex data transformations at granular levels—capabilities RSpec provides through flexible assertion syntax and comprehensive matcher libraries. The research emphasizes interoperability failures can trigger data loss, corruption, or misinterpretation, with potentially serious consequences for patient care and clinical research.

Applying RSpec to healthcare software testing addresses critical technical validation needs complementing workflow-focused testing Cucumber provides. Research documents multiple cases where seemingly minor computational errors led to significant patient safety risks, including incorrect medication dosage recommendations and failure to generate appropriate clinical alerts [7]. The granular nature of RSpec tests allows developers to verify clinical algorithms correctly handle comprehensive ranges of input values, including edge cases and exceptional situations difficult to model in higher-level acceptance tests. Similarly, research on interoperability challenges emphasizes healthcare data standards complexity and technical challenges tied to accurate data transformation between different systems and formats [8]. RSpec's flexible testing capabilities allow developers to verify data transformations maintain both structure and meaning of clinical information, ensuring critical patient data remains accurately preserved while moving between different systems and formats.

Testing Area	Complexity Level	Safety Impact	Detection Difficulty
Clinical Calculations	High	High	High
Algorithm Implementation	High	High	Medium
Exception Handling	Medium	High	High
Unit Conversion	Medium	High	Medium
HL7/FHIR Compliance	High	Medium	High
Semantic Data Integrity	High	High	High
Security Controls	Medium	High	Medium
Audit Logging	Medium	Medium	Low

Table 2: Critical Areas for RSpec Testing in Healthcare Applications [7, 8]

5. Patient Safety Scenario Testing

Among the most critical aspects of healthcare QA automation stands ensuring patient safety through comprehensive scenario testing. BDD helps in doing this as the QA engineers can model possible safety concerns and edge cases, simulate clinical situations that would cause harm to patients, confirm that the relevant protections and notifications are activated, and document tests that they do in case of regulatory audits.

Situations to test could be to verify allergy reminders, drug-to-drug interaction warnings or duplicate therapy warnings. As an example, in testing the drug-drug interaction alerts, the system should present suitable warnings where clinicians are about to prescribe medicines that are at risk of causing dangerous interactions with prescriptions that a patient already takes.

The implementation of comprehensive patient safety scenario testing represents a critical dimension of healthcare software quality assurance. Research analyzing clinical decision support systems and

Journal of Information Systems Engineering and Management
2025, 10(59s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

associated legal implications, highlight the complex balance between providing sufficient safety alerts and avoiding alert fatigue potentially causing clinicians to ignore critical warnings [9]. Their research examines how excessive or poorly implemented alerts can desensitize clinicians to warnings, potentially leading to missed critical safety notifications. The study emphasizes well-designed clinical decision support systems must carefully calibrate specificity and sensitivity of safety alerts, showing warnings only when truly warranted while ensuring critical safety issues never get missed. This analysis underscores thorough testing importance for alert mechanisms in healthcare applications, particularly for high-risk clinical areas such as medication ordering and administration. The research notes litigation concerns often drive healthcare organizations to implement overly sensitive alerting mechanisms, highlighting needs for testing methodologies validating both technical correctness of alerts and clinical appropriateness.

The value of scenario-based testing for patient safety gains further support from research examining computer-related patient safety incidents [10]. Their systematic analysis of 117 incidents from advanced hospital information systems identified several safety issue categories directly related to software functionality, including data input problems, data output/display errors, and system transfer issues. The research developed a comprehensive classification of computer-related safety incidents highlighting diverse ways software defects impact patient care. These findings emphasize many safety-critical issues involve complex interactions between system components, clinical workflows, and human factors, making them difficult to detect through isolated unit testing. The study notes approximately 55% of analyzed incidents carried potential for patient harm, with 45% requiring additional treatment or monitoring. These findings highlight critical importance of comprehensive scenario testing evaluating system behavior in realistic clinical situations, particularly for high-risk processes involving medication management and clinical decision support.

The structured approach BDD provides offers particular advantages for patient safety scenario testing in healthcare applications. While not specifically addressing BDD methodologies, research emphasizes importance of balancing safety alerts with clinical usability—a challenge demanding thorough testing of diverse clinical scenarios [9]. The research notes effective clinical decision support systems must correctly identify situations requiring alerts while avoiding excessive warnings contributing to alert fatigue. This balance requires testing beyond simple verification of alert triggers to encompass evaluation of alert appropriateness across diverse clinical scenarios. BDD's emphasis on explicit scenario definition provides an effective framework for this comprehensive testing approach. Similarly, classification of computer-related safety incidents highlights needs for testing methodologies addressing complex, multi-faceted nature of safety-critical scenarios in healthcare [10]. Their analysis identifies incidents spanning multiple categories, including data entry errors, system calculation issues, and data transfer problems. The high level workflow oriented test scenarios supported by BDD give a good model to develop and test such complex interactions in the clinical system where QA engineers would be able to check the correct behavior of the different systems with the different types of safety issues reported in the real life incident reports.

Safety Scenario Type	Occurrence (%)	Harm Potential	Detection Challenge	BDD Suitability
Alert System Effectiveness	High	High	Medium	High
Drug-Drug Interactions	High	High	Medium	High
Data Input Problems	Medium	High	High	Medium
Data Output/Display Errors	Medium	High	Medium	High
System Transfer Issues	Medium	Medium	High	Medium
Alert Fatigue Prevention	High	High	High	High
Calculation Accuracy	High	High	Medium	Medium
Allergy Verification	High	High	Low	High

Table 3: Critical Patient Safety Scenarios in Healthcare Software Testing [9, 10]

Journal of Information Systems Engineering and Management
2025, 10(59s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

6. Challenges and Best Practices

While BDD offers significant benefits for healthcare QA automation, several challenges demand attention: Test Data Management requiring creation of realistic test data covering diverse clinical scenarios while complying with privacy regulations; Performance Testing ensuring automated tests can evaluate system performance under realistic loads; Environment Complexity managing test environments accurately representing production configurations; and Integration with External Systems testing interfaces with laboratory, pharmacy, and other ancillary systems.

Best practices for addressing these challenges include implementing robust test data factories generating synthetic but clinically realistic data, using database cleaner strategies ensuring test isolation, leveraging continuous integration pipelines including security and performance testing, and implementing effective mocking strategies for external service dependencies.

Implementing effective healthcare QA automation requires addressing several significant challenges, particularly within test data management domains. Research through comprehensive analysis of hospital interoperability progress, highlight complex ecosystems of healthcare data exchange demanding accurate representation in testing environments [11]. Their research examining interoperability among 2,636 U.S. non-federal acute care hospitals found significant variability in data sharing capability implementation, with only 29.7 percent of hospitals engaging across all four measured interoperability domains: finding, sending, receiving, and integrating electronic patient information from outside providers. The study notes this heterogeneous landscape creates significant challenges for comprehensive testing, as healthcare applications must function correctly across diverse implementation patterns and varying external system capability levels. These findings emphasize effective testing for healthcare interoperability requires sophisticated approaches validating system behavior across complex, variable landscapes. While not directly addressing test data management, this research underscores challenges creating test scenarios accurately reflecting diversity of real-world interoperability implementations, highlighting needs for test data encompassing full ranges of exchange patterns observed in production environments.

Environment complexity and integration testing challenges in healthcare applications receive further exploration through research examining sociotechnical models for studying health information technology [12]. Their analysis presents an eight-dimensional model highlighting complex, interdependent nature of healthcare information systems, encompassing not just technical components but also human, organizational, and workflow factors. The study emphasizes healthcare applications exist within complex adaptive systems where changes in one dimension potentially trigger unpredictable effects across other dimensions. This research documents how such complexity creates significant challenges for comprehensive testing, as test environments must account for not just technical functionality but also sociotechnical contexts where systems operate. The research notes many healthcare IT failures stem from inadequate attention to sociotechnical interactions, highlighting importance of testing approaches addressing both technical correctness and sociotechnical integration. Their eight-dimensional model encompasses hardware and software computing infrastructure, clinical content, human-computer interface, people, workflow and communication, internal organizational policies, external rules and regulations, and system measurement and monitoring—all elements demanding consideration in comprehensive healthcare testing strategies.

Addressing these challenges requires implementing robust testing practices tailored to healthcare domains. While not specifically addressing BDD methodologies, research on interoperability progress provides context for understanding integration testing complexity across healthcare environments [11]. The research documents diverse interoperability landscapes across U.S. hospitals, highlighting needs for testing approaches validating system behavior across varying external system capability levels. This research underscores comprehensive testing strategy importance for external interfaces, including both technical validation of data exchange protocols and functional verification of clinical workflows spanning system boundaries. Similarly, sociotechnical model emphasizes multidimensional nature of healthcare IT implementations, highlighting needs for testing approaches addressing not just technical

functionality but also usability, workflow integration, and compliance with organizational and regulatory requirements [12]. Their research supports multi-layered testing strategy value combining technical validation with scenario-based testing of clinical workflows and organizational processes. The research emphasizes effective healthcare testing must consider complex interactions between technical systems and sociotechnical contexts, validating not just correct system functionality in isolation but also effective integration with broader healthcare environments.

Conclusion

BD provides a strong basis to focus on the complicated issues of health care software quality assurance. The combination of Cucumber to perform high level acceptance testing and RSpec to perform technical validation at the lower level can give a holistic system since they both cover clinical workflow and trials as well as technical precision. This twofold approach will assist healthcare organizations to successfully simulate complex clinical cases, qualify patient safety functionalities, ascertain regulatory requirements, and test interoperability between the organization and other system platforms. The interactive aspect of BDD, and the relative usability of the specifications by humans make it much easier to significantly engage the clinical stakeholders into the testing process which can result in more correct test scenarios and reveal the potential problems much earlier. Although adopting BDD in healthcare settings may include serious limitations that are associated with test data management, complexity of environments, and integration testing, all those challenges can be effectively solved with the use of strong technical practices and distinct approaches to methods. Since healthcare systems are becoming even more complex and vital, the BDD methodologies show a value of extreme merit in preserving quality, safety and dependability in this vital sphere.

References

[1] Jamiu O Busari, "Comparative analysis of quality assurance in health care delivery and higher medical education," Advances in Medical Education and Practice, 2012. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC3650879/
[2] Mohammad Z Anjum et al., "Development of Health Software using Behaviour Driven Development - BDD," In Proceedings of the 8th International Conference on Model-Driven Engineering and Software Development (MODELSWARD 2020), pages 149-157, 2022. [Online]. Available: https://www.scitepress.org/Papers/2020/89842/89842.pdf
[3] Abigail E Lewis et al., "Electronic health record data quality assessment and tools: a systematic review," Journal of the American Medical Informatics Association, 2023. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC10531113/
[4] Stefania Bruschi et al., "Excerpt from PNSQC Proceedings: Behavior Driven Development (BDD) A Case Study in Healthtech," ResearchGate, 2019. [Online]. Available: https://www.researchgate.net/publication/341434526
[5] Meghna Desai et al., "Implementation of Agile in healthcare: methodology for a multisite home hospital accelerator," BMJ Open Quality, 2024. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC11131107/
[6] C Tanner et al., "Electronic Health Records and Patient Safety," Applied Clinical Informatics, 2015. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC4377566/
[7] Adam Wright et al., "Analysis of clinical decision support system malfunctions: a case series and survey," 2016. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/27026616/
[8] Moritz Lehne et al., "Why digital medicine depends on interoperability," npj Digital Medicine, Volume 2, Article Number 79, 2019. [Online]. Available: https://www.nature.com/articles/s41746-019-0158-1
[9] Aaron S Kesselheim et al., "Clinical decision support systems could be modified to reduce 'alert fatigue' while still minimizing the risk of litigation," National Library of Medicine, 2011. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/22147858/

Journal of Information Systems Engineering and Management
2025, 10(59s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

[10] Farah Magrabi et al., "An analysis of computer-related patient safety incidents to inform the development of a classification," National Library of Medicine, 2010. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/20962128/
[11] A Jay Holmgren et al., "Progress In Interoperability: Measuring US Hospitals' Engagement In Sharing Patient Data," National Library of Medicine, 2017. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/28971929/
[12] Dean F Sittig and Hardeep Singh, "A new sociotechnical model for studying health information technology in complex adaptive healthcare systems," National Library of Medicine, 2010. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/20959322/