Taxonomy and Measurement Criteria for Automated Testing Frameworks in Healthcare Interoperability

Abstract

The digitization of healthcare has resulted in exponential data growth and integration of diverse systems, such as EHRs, imaging platforms, and IoT devices, but challenges like fragmented data, inconsistent standards, and frequent system updates hinder interoperability and threaten patient safety and operational efficiency. Automated testing has become essential for ensuring system reliability, yet finite resources and varying API criticality demand a strategic approach. This paper proposes a framework comprising a taxonomy of automated testing strategies, healthcare-specific evaluation criteria, and a risk assessment model to prioritize high-impact API regions. By focusing on critical areas such as clinical workflows, regulatory compliance, and data sensitivity, the framework provides actionable tools for maximizing the results of deploying automated testing methodologies. The result demonstrates the effectiveness of functional validation for safety-critical workflows and highlight the inefficacy of methods like chaos testing in healthcare. The findings provide a structured approach to evaluating automated testing methodologies, offering insights into how different strategies align with the unique demands of healthcare interoperability, such as ensuring system resilience, regulatory compliance, and data accuracy. The framework's applicability is supported by examples of prioritizing high-risk API regions and is positioned for future validation through real-world case studies and standardization efforts to advance healthcare interoperability.

Introduction

The increasing digitization of healthcare has driven the rapid integration of diverse technologies, such as electronic health records (EHRs), wearable devices, and imaging systems, leading to an exponential growth in healthcare data. By 2020, healthcare data accounted for 2.3 zettabytes globally, with an annual growth rate of 36%—surpassing most other industries ([1] Moore; [2] RBC Capital Markets). While this technological evolution holds a lot of potential to enhance patient outcomes and clinical efficiency, it also introduces critical challenges, such as fragmented data, inconsistent standards, and increasingly complex workflows. These barriers hinder the seamless interoperability essential for effective data exchange across healthcare systems.

With these challenges, automated testing has is a potential tool for validating interoperability and ensuring the reliability of interconnected systems through numerous version upgrades throughout the complex healthcare technology ecosystem. However, the finite nature of resources at the disposal of health systems and the variability in the importance of different API regions necessitate a strategic approach to testing. This paper addresses the research question:

How can healthcare organizations strategically allocate limited resources to select and deploy the most effective automated testing methodologies for maximizing impact?

The objectives of this paper are to:

1. Propose a taxonomy of automated testing strategies tailored to healthcare interoperability.

2. Introduce healthcare-specific measurement criteria to evaluate the effectiveness of testing methodologies.
3. Develop a risk assessment model to prioritize testing efforts by identifying high-impact API regions.

The scope of this research encompasses the evaluation of automated testing strategies within healthcare interoperability systems, focusing on their alignment with clinical workflows, regulatory compliance, and operational efficiency. This study draws on industry practices, trade literature, and theoretical constructs to address the unique demands of healthcare systems.

This paper is organized as follows:

- **Background and Literature Review** establish the foundational concepts of interoperability and highlight existing challenges.
- **Current Automated Testing Methods for Healthcare Interoperability** enumerates various automated testing methodologies as a foundation for the taxonomy, risk assessment, and criteria creation.
- **Taxonomy of Automated Testing Strategies** categorizes testing methodologies, while the proposed healthcare-specific measurement criteria ensure these methods address domain-specific requirements.
- **Healthcare-Specific Measurement Criteria for Automated Testing** section identifies criteria that can evaluate and distinguish between various automated testing strategies within a taxonomic category.
- **Risk Assessment of API to Downstream Workflows** prioritizes testing efforts, offering a systematic approach to resource allocation.
- **Result: Example of Application and Validation** gives a concrete and practical example of how the system should be used, generating a clear interpretable result.
- **Discussion** section synthesizes these contributions, addressing their implications and limitations.
- **Conclusion** summarizes the findings, contributions, and areas for future research.

Through this structured approach, the paper aims to provide actionable insights for optimizing automated testing efforts in healthcare interoperability.

Background and Literature Review

Healthcare interoperability refers to the ability of diverse healthcare systems, software applications, and devices to communicate and exchange data seamlessly. It is a cornerstone of modern healthcare, enabling the integration of patient information across various platforms such as electronic health records (EHRs), medical imaging systems, and real-time monitoring devices. Appendix A shows a partial example of This capability ensures that clinicians have access to accurate and comprehensive data, regardless of where care is provided, which is essential for informed decision-making and efficient patient care.

Interoperability operates at three levels:

- **Foundational Interoperability:** This basic level allows healthcare systems to send and receive data but does not ensure that the receiving system can interpret the data.
- **Structural Interoperability:** At this level, data exchanges between systems are structured using standardized formats, such as HL7 or FHIR. This ensures that data is stored and displayed consistently across systems.
- **Semantic Interoperability:** The most advanced level of interoperability, where systems not only exchange data but also interpret it accurately. This enables actionable insights by ensuring consistent understanding of medical terminology, codes, and data contexts across platforms.

Several standards exist in healthcare interoperability each focused on different content and structure of communication. The most used interface standards are:

- **HL7 (Health Level 7):** Focuses on clinical, administrative, and financial data exchange, enabling compatibility across systems.
- **FHIR (Fast Healthcare Interoperability Resources):** A modern standard built on RESTful APIs, facilitating web-based data exchange by breaking healthcare information into discrete "resources" like patients or medications.
- **DICOM (Digital Imaging and Communications in Medicine):** Standardizes the handling, storage, and exchange of medical images, ensuring seamless communication between imaging systems and EHRs.

As healthcare systems evolve, the push towards semantic interoperability is increasingly critical. It ensures not just data exchange but also meaningful data interpretation, allowing systems to harmonize medical terminologies and integrate clinical data effectively. This level of integration is essential for enabling advanced applications such as clinical decision support tools, personalized medicine, and population health management.

The literature shows healthcare interoperability faces two significant challenges: data fragmentation and the complexity of workflows.

Fragmentation of Data: Healthcare data is often scattered across various platforms, including EHRs, medical imaging systems, and IoT devices. Much of this data is unstructured and stored in different formats, making it difficult to integrate and analyze. This lack of standardization hinders the seamless exchange of information, resulting in inefficiencies and limiting the ability to leverage data for clinical decision-making. Studies have highlighted that persistent data quality issues and fragmentation remain significant barriers to achieving interoperability in healthcare systems ([3] D'Amore et al., 2018; [4] Gurumurthy et al., 2020). Semantic technologies, such as knowledge graphs, have been proposed as a solution to integrate fragmented data and improve its usability for clinical care ([5] Bashir et al., 2022). Additionally, modeling frameworks like ontologies have been developed to address inconsistencies and enable seamless data exchange between systems ([6] Osorio et al., 2022). Recommendations for achieving shareable and interoperable data further emphasize the need for cohesive integration to overcome fragmentation ([7] Szarfman et al., 2022).

Complexity of Workflows: Interoperable systems must support intricate healthcare workflows that involve multiple stakeholders and systems. Frequent updates, such as quarterly EHR releases or biweekly API changes, exacerbate the risk of downstream failures. Even minor disruptions in workflows can lead to delays in care, increased clinician burden, and potential patient harm, emphasizing the need for robust testing and validation processes. Poor interoperability can amplify these challenges, as fragmented workflows disrupt clinical efficiency and increase the risk of errors ([8] Walker et al., 2023; [3] D'Amore et al., 2018). Studies highlight the critical impact of EHR interoperability on clinical safety and quality of care, demonstrating how workflow complexities can hinder timely decision-making ([9] Li et al., 2022). Additionally, ensuring that system changes do not interfere with existing workflows requires targeted testing approaches, such as regulatory-driven automated test suites ([10] Morrison et al., 2013) and automated document transformation testing (X] Pascale et al., 2009). Recommendations for improving interoperability also stress the importance of workflow alignment to reduce disruptions and enhance clinical outcomes ([7] Szarfman et al., 2022).

The risks associated with changes within a healthcare ecosystem increase as interoperability deepens, with frequent updates—such as quarterly EHR upgrades or biweekly scrum releases—heightening the potential for downstream failures. These failures can disrupt workflows, delay care, and compromise patient safety, making system reliability increasingly difficult to maintain. Manual testing cannot keep pace with the growing complexity and scale of these interconnected systems. As a result, rigorous and efficient automated testing by healthcare analysts is critical to minimizing these risks, ensuring seamless data exchange, and maintaining the reliability of the ecosystem.

Automated testing enables healthcare systems to validate updates and integrations efficiently, ensuring that changes do not introduce errors or disrupt interoperability. By providing consistent, repeatable, and scalable testing processes, automation reduces the likelihood of errors going undetected and allows for thorough validation across the interconnected ecosystem ([10] Morrison et al., 2013; [11] Pascale et al., 2009). This approach is particularly critical for systems reliant on standards like FHIR, HL7, and DICOM, where even minor misconfigurations can propagate through workflows and impact clinical outcomes. Automated frameworks, such as those designed for RESTful APIs, streamline testing processes and improve fault detection rates, ensuring the resilience of API-driven systems ([12] Lei et al., 2023; [13] Ehsan et al., 2022).

In addition to improving efficiency, automated testing plays a key role in enhancing system resilience and safety. Misconfigured APIs, for example, can lead to incomplete or incorrect data exchanges, potentially resulting in delays or errors in patient care. Automated tests systematically validate these APIs and other system components, identifying vulnerabilities and ensuring that data flows remain accurate and consistent ([14] Ahmed et al., 2023; [15] Segura et al., 2018). For healthcare-specific use cases, test suites tailored to regulatory compliance further ensure stability and adherence to critical standards ([10] Morrison et al., 2013). By addressing the challenges of fragmentation, complexity, and frequent updates, automated testing ensures that healthcare systems can continue to evolve without compromising interoperability, efficiency, or safety ([16] Vega et al., 2010).

Automated testing is a powerful but limited resource, requiring careful planning and maintenance to ensure its effectiveness. By developing a taxonomy of automated testing strategies and risk measures for APIs, this paper provides healthcare technical analysts with tools to optimize their testing efforts. A robust taxonomy will help analysts identify which aspects of a complex health system require validation, ensuring that the critical areas most likely to impact patient safety, security, and operational revenue are prioritized. Simultaneously, risk measures will enable analysts to deploy automated tests strategically, focusing on areas where failures could have the most severe consequences, such as clinical workflows, data exchange processes, and compliance with regulatory standards. Together, these tools minimize unnecessary testing expenses while maximizing the impact of automated tests, ensuring that healthcare systems remain resilient, efficient, and safe in an ever-evolving technological landscape.

Current Automated Testing Methods for Healthcare Interoperability

Healthcare and other industries employ several methods of testing that can be automated to perform interoperability testing, and it is anticipated that new methods will emerge as semantic interoperability continues to evolve. These methods, outlined in Table 1, are described to support the development of a taxonomy and measurement criteria, enabling the evaluation of both current and future testing approaches. No scholarly studies have been identified that comprehensively enumerate methods for interoperability testing, and so this list was generated as an amalgamation of automated testing strategies for interoperability across a variety of industries and was partially constructed with the help of ChatGPT. It is intended to be illustrative and is not intended to be a comprehensive list.

Table 1: List and description of various automated testing approaches for interoperability

Type	Purpose	Examples	Tools
Functional Testing	Validates that system functionalities work as intended and meet specified requirements.	Ensuring an API correctly retrieves and processes patient demographics; Verifying that a lab result is properly transmitted from the lab system to the EHR.	Inferno, Postman
Integration Testing	Validates interactions between multiple systems or components to ensure data flows correctly across interfaces.	Testing data exchange between an EHR and a pharmacy system using HL7 or FHIR; Validating that imaging data from a PACS system is accessible in an EHR via DICOM.	SoapUI, MuleSoft
Performance Testing	Ensures that the system performs efficiently under expected workloads, such as high transaction volumes or concurrent users.	Testing the response time of a FHIR API under heavy load; Evaluating the scalability of a health information exchange (HIE) during peak usage.	JMeter, Gatling

Security Testing	Identifies vulnerabilities and ensures the security of data exchanges and system access points.	Validating compliance with HIPAA for patient data encryption during transfers; Detecting vulnerabilities in APIs that could lead to unauthorized access or data breaches.	OWASP ZAP, Burp Suite
Semantic Validation	Ensures that data is exchanged and interpreted consistently across systems.	Verifying that a medication allergy coded in one system is correctly understood by another; Ensuring proper mapping of terminology standards like SNOMED CT or LOINC.	Custom validation scripts, ontological testing tools
Regression Testing	Ensures that new updates or changes do not break existing functionality.	Testing that a quarterly EHR upgrade does not disrupt existing API connections; Verifying that downstream workflows are unaffected by new FHIR resource definitions.	Selenium, CI/CD pipelines with automated test scripts
Load and Stress Testing	Evaluates system behavior under extreme conditions or beyond normal operational capacity.	Simulating thousands of concurrent FHIR API requests to test system resilience; Stress-testing an HIE for large-scale data transactions during a regional health emergency.	Apache JMeter, LoadRunner
End-to-End Testing	Validates the entire workflow, ensuring that data flows accurately and reliably across all systems in a process.	Testing a patient admission workflow from the scheduling system to the billing system; Ensuring seamless integration of data from wearable devices into EHRs.	Test automation frameworks customized for healthcare workflows
Compliance Testing	Verifies that systems meet regulatory and interoperability standards, such as HIPAA or ONC certification requirements.	Testing FHIR API conformance to implementation guides; Ensuring that an EHR system complies with ONCs interoperability mandates.	Inferno, NIST Conformance Testing Tools
Interoperability-Specific Testing	Focuses on validating standards and protocols unique to healthcare interoperability.	Testing the compliance of HL7 message exchanges; Ensuring accurate medical image transfers using DICOM standards.	IHE Gazelle, FHIR Validator
Scenario-Based Testing	Evaluates how systems perform under specific clinical or operational scenarios.	Testing how a telehealth system integrates with EHRs during a virtual visit; Simulating an emergency care scenario to	Custom-built test scenarios, simulation environments

		validate data sharing between EMS and hospitals.	
Chaos Testing	Chaos testing intentionally introduces faults or failures into a system to evaluate its resilience and ability to recover.	Simulating server outages, network disruptions, or database failures to observe system behavior and recovery mechanisms.	Chaos Monkey, Gremlin, Pumba

Taxonomy of Automated Testing Strategies

Healthcare interoperability testing encompasses a diverse set of strategies designed to ensure the reliability, functionality, and security of interconnected systems. These strategies can be categorized into six key groups based on their focus and objectives. Table 2 provides an overview of this taxonomy, highlighting the testing types under each category and their purposes.

Functional Validation focuses on verifying the correctness of individual system components and maintaining their functionality after updates or changes. Functional testing ensures that specific modules, such as an Ambulatory Clinical Workflow Module (Epic EpicCare for example), retrieve and process data accurately, while regression testing validates that modifications do not disrupt existing workflows.

Interface and Data Exchange Testing emphasizes the validation of interactions between multiple systems, ensuring seamless data exchange and integration. Integration testing evaluates connections between EHRs, imaging systems, and other platforms, while interoperability-specific testing focuses on standards like HL7 and DICOM to ensure compliance and compatibility.

Performance and Reliability Testing assesses system behavior under different workloads to ensure scalability and robustness. Performance testing validates the system's efficiency during normal operations, and load and stress testing evaluates resilience under extreme conditions, such as high transaction volumes during peak usage.

Semantic Validation and Compliance address healthcare-specific requirements for data consistency, accurate interpretation, and adherence to standards. Semantic validation ensures that data exchanged between systems retains its meaning and usability, while compliance testing verifies conformity to regulations like HIPAA and ONC mandates.

Security and Risk Mitigation Testing safeguards sensitive healthcare data by identifying vulnerabilities and assessing potential risks. Security testing focuses on detecting threat risks or security holes such as a point where unauthorized access is possible, or where data breaches could occur, while incorporating risk measures helps prioritize testing efforts on critical areas.

Workflow and Scenario-Based Testing validates end-to-end workflows and specific operational scenarios. End-to-end testing ensures that data flows smoothly across all system

components in a clinical process, and scenario-based testing evaluates system performance in real-world settings, such as telehealth consultations or emergency care scenarios.

These categories collectively provide a framework for deploying automated testing strategies effectively, ensuring that healthcare systems remain resilient, efficient, and safe. By addressing the unique challenges of healthcare interoperability, this taxonomy aids technical analysts in optimizing testing resources for maximal impact.

Table 2. Taxonomy of Healthcare Automated Interoperability Testing

Category	Testing Types	Purpose
Functional Validation	Functional Testing, Regression Testing, Chaos Testing	Ensure individual components work correctly and maintain functionality after changes.
Interface and Data Exchange	Integration Testing, Interoperability-Specific Testing	Validate the integration and data exchange between systems.
Performance and Reliability	Performance Testing, Load and Stress Testing	Assess system scalability, reliability, and behavior under heavy workloads.
Semantic Validation & Compliance	Semantic Validation, Compliance Testing	Ensure data consistency and adherence to interoperability standards.
Security and Risk Mitigation	Security Testing	Identify and address vulnerabilities, ensuring data security and minimizing risks.
Workflow and Scenario-Based	End-to-End Testing, Scenario-Based Testing	Validate real-world clinical workflows and specific operational scenarios.

Healthcare-Specific Measurement Criteria for Automated Testing

Evaluation criteria for testing have been well established since the early days of software development, with measures such as Coverage Criteria, Fault Detection Rate, and Test Reliability being well understood. These metrics ensure that testing strategies can be compared objectively. However, healthcare systems present unique challenges that require additional, domain-specific measurements to adequately evaluate the effectiveness of automated testing strategies. This paper proposes the following measures to be considered when evaluating automated testing techniques against each other. Table 4 shows the proposed healthcare-specific criteria that complement traditional measures.

These healthcare-specific criteria address the unique needs of healthcare interoperability, emphasizing patient safety, regulatory compliance, and seamless workflow integration. By incorporating these measures, testing strategies can be better evaluated for their ability to meet the complex demands of healthcare systems while ensuring reliability, security, and efficiency.

Table 4. Healthcare-Specific Measurement Criteria for Automated Testing

Criterion	Description	Measurement Guideline
Clinical Workflow Accuracy	Tests must validate critical healthcare workflows, such as medication ordering, lab result reporting, or patient admissions. Accurate testing of these workflows ensures system reliability and prevents disruptions that could harm patients.	Boolean (Pass/Fail): Does the test accurately simulate and validate the clinical workflow?
Interoperability Compliance	Testing strategies must verify adherence to healthcare interoperability standards like HL7, FHIR, DICOM, and C-CDA. This ensures seamless communication between systems and eliminates inconsistencies in data exchange	Boolean (Pass/Fail): Does the test ensure adherence to standards (e.g., HL7, FHIR, DICOM)?
Patient Safety Impact	Testing strategies should prioritize areas where errors could directly affect patient safety, such as medication administration or allergy data. The ability to identify safety-critical issues is essential for ensuring high-quality care.	Boolean (Pass/Fail): Does the test detect errors that could harm patients?
Data Integrity Validation	Tests must confirm that data exchanged between systems is complete, accurate, and unaltered. This is crucial for maintaining trust in clinical decisions based on shared data.	Boolean (Pass/Fail): Is the data exchanged complete, accurate, and unaltered after transmission?
Semantic Consistency	Systems must interpret and use data uniformly across platforms. Tests should evaluate the consistency of medical terminologies and data representations, ensuring semantic interoperability.	Boolean (Pass/Fail): Does the test ensure consistent interpretation of data across systems?
Regulatory Compliance Assurance	Testing strategies must ensure alignment with healthcare regulations, such as HIPAA and ONC requirements. This minimizes legal and operational risks associated with non-compliance.	Boolean (Pass/Fail): Does the test validate compliance with healthcare regulations (e.g., HIPAA, ONC)?
Real-Time Testing Capability	The ability to test APIs and workflows in real-time scenarios is critical for applications like emergency care and remote monitoring. Real-time validation ensures system reliability under time-sensitive conditions.	Time Measure (ms): What is the response time for API validation and data exchange during real-time scenarios?

Error Propagation Risk Assessment	Testing should assess how errors in one system or API region may propagate across downstream workflows, enabling targeted testing of high-risk areas.	Scale (1-3): Evaluate the potential downstream impact of an error. High Risk (3), Medium Risk (2), Low Risk (1).
Healthcare Data Sensitivity	Automated tests must evaluate the secure handling of sensitive patient data, such as diagnoses, lab results, and treatment plans, to safeguard privacy and compliance.	Boolean (Pass/Fail): Does the test safeguard sensitive data, such as encryption and secure transmission?
Impact on Operational Efficiency	Testing must also consider the impact on end-user workflows. Effective strategies reduce clinician burden by ensuring smooth interoperability, which improves care delivery and satisfaction.	Time Measure (minutes): Measure the reduction in time or clicks required for clinicians due to smoother interoperability.

Risk Assessment of API to Downstream Workflows

As healthcare systems evolve towards deeper semantic interoperability, APIs have become more prevalent as the mechanism for connecting disparate applications, enabling the seamless exchange of critical data. However, not all APIs—and even not all regions within an API—carry the same level of risk. Some handle highly sensitive, safety-critical information like medication orders or allergy data, while others manage less impactful tasks, such as room assignments or timestamps. Given the constraints of time and resources, it is important to assess the relative risk of each API and its components to ensure automated testing efforts are focused where they are needed most.

APIs often influence decisions and actions across clinical, administrative, and operational domains. For example, the failure of an API responsible for transmitting medication orders could lead to errors in treatment, jeopardizing patient safety. Similarly, disruptions in billing-related APIs could result in significant financial consequences for healthcare organizations. Conversely, errors in less critical areas, such as room assignments, might have negligible impact and may not justify extensive automated testing.

Assessing API risk provides a systematic approach to prioritizing automated testing efforts, enabling developers and healthcare analysts to allocate resources efficiently by evaluating factors such as patient safety impact, regulatory compliance requirements, and the potential for error propagation. High-risk areas, such as those related to clinical decision-making or billing, can be targeted for rigorous testing, while lower-risk areas receive fewer resources. A robust risk assessment framework ensures that limited testing efforts yield the greatest possible impact. Automated testing is resource-intensive, requiring significant effort to develop and maintain. By focusing on API regions with the highest potential to disrupt workflows or harm patients, risk assessment helps balance testing costs with benefits.

Risk assessment also informs the strategic deployment of testing within complex healthcare ecosystems. By identifying high-risk areas that could propagate errors across downstream workflows, developers can ensure that automated tests are applied where they will have the maximum impact. This targeted approach not only improves the reliability and resilience of interconnected systems but also safeguards patient safety, protects financial integrity, and enhances operational efficiency. ***Table 3*** shows the various risk criteria developed for this paper, designed in the image of similar subjective mental health risk evaluations like PHQ-9 [17] which are common throughout the healthcare industry and familiar to the health systems who would use this risk assessment strategy.

Formula for Risk Score

The formula for Risk Score aggregates the weighted scores of the criteria to determine the API region's overall risk level.

Risk Score=(PSI×1.5)+(RC×1)+(DS×1.5)+(FoU×1)+(WC×1.5)+(FI×1.5)+(ID×1)+(EPR×1)+(HS×0.5)+(CF×1.5)+(DU×0.5)+(SN×0.5)

Weights: Higher weights (e.g., 1.5) are assigned to criteria like patient safety and workflow criticality to emphasize their importance. Lower weights (e.g., 0.5) are assigned to criteria like historical stability to reflect their lesser impact on risk.

Risk Categories:

- **High Risk:** 31+
- **Medium Risk:** 22–30
- **Low Risk:** <22

Table 3. API Risk Assessment Criteria

Criterion	Description	Scoring
Patient Safety Impact (PSI)	Does this API region handle information directly related to patient safety, such as medications, allergies, or orders?	High (3) for safety-critical data, Medium (2) for indirectly related, Low (1) for unrelated.
Regulatory Compliance (RC)	Is the data governed by strict compliance requirements such as HIPAA or ONC regulations?	High (3) if compliance-critical, Medium (2) for general clinical data, Low (1) for metadata or auxiliary data.
Data Sensitivity (DS)	Does the API handle sensitive patient information such as diagnoses, lab results, or health conditions?	High (3) for highly sensitive, Medium (2) for moderately sensitive, Low (1) for non-sensitive data.
Frequency of Use (FoU)	How often is this API called in typical operations?	High (3) for frequent calls, Medium (2) for occasional calls, Low (1) for infrequent calls.

Workflow Criticality (WC)	Does the API impact critical workflows like ordering, billing, or care delivery?	High (3) for critical workflows, Medium (2) for secondary workflows, Low (1) for ancillary tasks.
Financial Impact (FI)	Does the API directly affect revenue streams, such as billing or insurance data?	High (3) for financial-critical data, Medium (2) for moderate impact, Low (1) for no financial impact.
Interoperability Dependency (ID)	Does the API rely on or feed into other systems, requiring seamless data exchange?	High (3) for APIs with many dependencies, Medium (2) for some dependencies, Low (1) for standalone APIs.
Error Propagation Risk (EPR)	If an issue arises, how far-reaching are the downstream effects?	High (3) for widespread effects, Medium (2) for localized effects, Low (1) for minimal effects.
Historical Stability (HS)	How frequently have issues or updates occurred in this API region?	High (1) for frequent issues, Medium (2) for occasional issues, Low (3) for historically stable.
Change Frequency (CF)	How often is this API or the underlying data updated or modified?	High (3) for frequent updates, Medium (2) for occasional updates, Low (1) for stable regions.
Developer Understanding (DU)	How complex is the API region, and how well do developers understand its function?	High (1) for unclear or complex regions, Medium (2) for moderate understanding, Low (3) for well-understood regions.
Superfluous Nature (SN)	Does the API handle auxiliary or non-critical data like room assignments or timestamps?	High (1) for non-critical data, Medium (2) for moderately critical, Low (3) for critical data.

Result: Example of Application and Validation

The example comparison in Table 5 shows the differing applicability of Functional Testing and Chaos Testing within the context of healthcare interoperability as functional validation of the healthcare technology ecosystem. Each criterion highlights a specific aspect of testing, such as clinical workflow accuracy, regulatory compliance, and patient safety impact, to determine the suitability of these strategies in addressing healthcare-specific challenges. For instance, functional testing excels in areas like clinical workflow accuracy, ensuring that critical functions like medication orders and lab results operate as intended. Conversely, chaos testing, while valuable for stress-testing system resilience, demonstrates low applicability to compliance and patient safety due to its disruptive nature.

This comparison demonstrates the importance of aligning testing strategies with the unique demands of healthcare. For example, functional testing emerges as a "good candidate" for healthcare settings, as it directly validates key functionalities that underpin patient care. In contrast, chaos testing, with its focus on fault injection and system resilience, is deemed a "poor

candidate" for healthcare due to its inability to address safety-critical or compliance-focused scenarios and the inherent risk chaos testing injects into the technology ecosystem.

When interpreting Table 6, which assesses API risk by region, the evaluation emphasizes how different API functionalities contribute to overall risk. For instance, high-risk regions like Allergies and Orders Placed align closely with the areas where functional testing is most effective, highlighting the need for targeted validation to ensure patient safety and workflow integrity. Low-risk regions like Room Assignment or Patient Arrival Time might not warrant the same level of scrutiny.

Reading these tables together would guide healthcare analysts to invest heavily in automated functional testing in the allergies and orders areas of the API. If funding and time allows, analysts could extend automated testing into the patient arrival and room assignment areas of the API, or they could refocus their efforts to higher impact projects. Additionally, this analysis would show healthcare analysts that Chaos Testing should be rejected as a viable method of testing in a healthcare environment.

Table 5. Example comparison of Functional Testing and Chaos Testing

Criterion	Functional Testing	Chaos Testing
Category	Functional Validation	Functional Validation
Clinical Workflow Accuracy	Pass: Validates specific functionality critical to clinical workflows, such as order placements and lab result retrieval.	Fail: Chaos testing focuses on system resilience rather than specific workflow accuracy, which could disrupt workflows.
Interoperability Compliance	Pass: Ensures that individual APIs and components adhere to interoperability standards (e.g., FHIR, HL7, DICOM).	Fail: Chaos testing does not explicitly validate compliance with interoperability standards.
Patient Safety Impact	Pass: Directly tests safety-critical functionalities (e.g., allergy alerts, medication orders).	Fail: Deliberate faults could inadvertently disrupt safety-critical systems, making this unsuitable for live systems.
Data Integrity Validation	Pass: Ensures that inputs and outputs of individual components meet healthcare data integrity requirements.	Fail: Can reveal data corruption during failures, but this is not its primary focus.
Semantic Consistency	Pass: Limited to validating that data at the component level adheres to expected semantics.	Fail: Chaos testing does not focus on ensuring semantic consistency.
Regulatory Compliance Assurance	Pass: Validates individual modules for compliance with regulatory standards like HIPAA or GDPR.	Fail: Chaos testing typically overlooks compliance-specific validations.
Real-Time Testing Capability	1 - Low: Typically does not simulate real-time or high-load scenarios but focuses on individual functionality.	3 - High: Tests system resilience under real-time failures, providing insights into real-world behavior under stress.
Error Propagation Risk Assessment	1 - Low: Focuses on isolated components, with limited ability to assess downstream effects of errors.	2 - Medium: Chaos testing reveals how failures propagate across interconnected systems.

Healthcare Data Sensitivity	Pass: Ensures sensitive data (e.g., diagnoses, medications) is processed securely.	Fail: Fault injections could expose sensitive data during tests, posing security risks.
Impact on Operational Efficiency	1 - Low: Focuses on individual components, offering limited insights into overall operational efficiency.	1 - Low: Can identify systemic inefficiencies during failure recovery, but it's not its primary purpose.
Evaluation	**Good candidate for use in healthcare setting**	**Poor candidate for use in healthcare setting**

Table 6. Example API Risk Assessment

API Region	PSI	RC	DS	FoU	WC	FI	ID	EPR	HS	CF	DU	SN	Risk Score	Category
Allergies	3	3	3	3	3	2	3	3	2	2	1	1	34	High Risk
Room Assignment	1	1	1	2	1	1	1	1	3	1	3	3	17	Low Risk
Orders Placed	3	3	3	2	3	3	3	3	2	2	2	1	34.5	High Risk
Patient Arrival Time	1	1	1	2	1	1	1	1	3	1	3	3	17	Low Risk

Discussion

This paper provides a structured framework for evaluating automated testing methods in healthcare interoperability, introducing a taxonomy for testing strategies, healthcare-specific evaluation criteria, and a risk assessment model to prioritize testing efforts. These tools collectively address the critical question of how healthcare systems can maximize the impact of automated testing given finite resources and time. The findings emphasize that not all automated testing strategies are equally effective in healthcare, where safety-critical workflows and compliance-sensitive systems demand higher scrutiny. The proposed API risk assessment ensures testing resources are directed toward areas with the greatest potential impact, such as safeguarding patient outcomes and operational integrity.

The contributions of this work build on existing software testing practices by extending traditional metrics—such as fault detection rate and coverage criteria—with healthcare-specific dimensions like patient safety impact, regulatory compliance assurance, and data sensitivity validation. These additions address the unique demands of healthcare systems, where even minor errors can have far-reaching consequences for clinical care and operational efficiency. The integration of a risk assessment framework complements existing interoperability research by offering a practical mechanism to allocate testing resources effectively.

A key limitation of this approach is that the proposed taxonomy and criteria are theoretical constructs supported by industry practices and trade literature in the absence of empirical studies. Additionally, the subjective nature of the risk assessment framework introduces variability in implementation. Future research should focus on validating the proposed framework through real-world case studies and exploring the application of machine learning techniques to enhance the accuracy and efficiency of risk scoring for API regions. Further development of standardized tools to

measure healthcare-specific testing criteria would also facilitate consistent and reliable evaluations.

By addressing these critical challenges, this paper equips healthcare organizations with a practical approach to maximizing the return of automated testing efforts. The findings have implications for healthcare organizations striving to balance finite resources with the need for robust, reliable systems. This work contributes to advancing safer, more efficient healthcare interoperability, supporting the broader goal of improving patient care and technology ecosystem resilience.

Conclusion

This paper addresses the question: *How can healthcare organizations strategically allocate limited resources to select and deploy the most effective automated testing methodologies for maximizing impact?* In response, it proposes a comprehensive framework comprising a taxonomy of testing strategies, healthcare-specific evaluation criteria, and a risk assessment model for API prioritization. Together, these tools provide healthcare analysts with a practical approach to evaluating and deploying automated testing efforts in an environment where patient safety, regulatory compliance, and operational efficiency are paramount.

The findings of this study emphasize that not all automated testing methodologies are equally suited for healthcare interoperability. By incorporating healthcare-specific criteria—such as clinical workflow accuracy, patient safety impact, and regulatory compliance assurance—the framework ensures that testing strategies align with the challenges and priorities of healthcare systems. The risk assessment model complements this approach by enabling the prioritization of high-impact API regions, ensuring that testing resources are focused on areas with the greatest potential for improving patient outcomes and minimizing operational risks.

This work contributes to the field by extending traditional software testing evaluation methods with domain-specific considerations for healthcare. It highlights the importance of targeted testing investments to maximize the effectiveness of finite resources. While the proposed framework is theoretical, its practical application offers significant promise for improving the resilience, safety, and efficiency of healthcare technology ecosystems.

Future research should focus on validating the proposed models through real-world case studies and developing standardized tools to measure healthcare-specific criteria more reliably. Advancements in machine learning and data analytics could further enhance the precision of API risk scoring, allowing for health systems to adapt to APIs that increase in complexity over releases. By addressing these areas, this research lays the foundation for more robust and impactful automated testing frameworks in healthcare, supporting the broader goal of achieving seamless and reliable interoperability that remains robust through the frequent agile upgrades to applications within the healthcare technology ecosystem.

References and Acknowledgement

This paper utilized ***ChatGPT*** for assistance in refining structure and enhancing clarity in commentary. All original ideas, analyses, and conclusions presented are those of the author.

[1] J. Moore, Yasmin Dias Guichot, and World Economic Forum, "How to harness health data to improve patient outcomes," World Economic Forum, Jan. 05, 2024. https://www.weforum.org/agenda/2024/01/how-to-harness-health-data-to-improve-patient-outcomes-wef24/

[2]. "RBC Capital Markets | Navigating the changing Face of healthcare episode." https://www.rbccm.com/en/gib/healthcare/episode/the_healthcare_data_explosion

[3] J. D'Amore et al., "Interoperability Progress and Remaining Data Quality Barriers of Certified Health Information Technologies," AMIA Annu Symp Proc, vol. 2018, pp. 358–367, Dec. 2018.

[4] R. Gurumurthy, D. Schatsky, and J. Camhi, "2020 Digital transformation survey | Deloitte Insights," www2.deloitte.com, May 26, 2020. https://www2.deloitte.com/us/en/insights/topics/digital-transformation/digital-transformation-survey.html

[5] A. Bashir, R. Nagpal, D. Mehrotra, and M. Bala, "Interoperability of Electronic Health Records for Dyslipidemia using Knowledge graphs," in 2023 13th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Jan. 2023, pp. 124–129. doi: 10.1109/Confluence56041.2023.10048874.

[6] V. Osorio, S. Laviosa, G. Giménez-Lugo, and C. Villalba, "Data Modeling with Ontological Formalism for Semantic Interoperability between Health Information Systems, applied to Primary Health Care in Paraguay," in 2022 XVLIII Latin American Computer Conference (CLEI), Oct. 2022, pp. 1–10. doi: 10.1109/CLEI56649.2022.9959960.

[7] A. Szarfman et al., "Recommendations for achieving interoperable and shareable medical data in the USA," Commun Med, vol. 2, no. 1, pp. 1–7, Jul. 2022, doi: 10.1038/s43856-022-00148-x.

[8] D. M. Walker, W. L. Tarver, P. Jonnalagadda, L. Ranbom, E. W. Ford, and S. Rahurkar, "Perspectives on Challenges and Opportunities for Interoperability: Findings From Key Informant Interviews With Stakeholders in Ohio," JMIR Med Inform, vol. 11, p. e43848, Feb. 2023, doi: 10.2196/43848.

[9] E. Li, J. Clarke, H. Ashrafian, A. Darzi, and A. L. Neves, "The Impact of Electronic Health Record Interoperability on Safety and Quality of Care in High-Income Countries: Systematic Review," Journal of Medical Internet Research, vol. 24, no. 9, p. e38144, Sep. 2022, doi: 10.2196/38144.

[10] P. Morrison, C. Holmgreen, A. Massey, and L. Williams, "Proposing regulatory-driven automated test suites for electronic health record systems," in 2013 5th International Workshop on Software Engineering in Health Care (SEHC), May 2013, pp. 46–49. doi: 10.1109/SEHC.2013.6602477.

[11] M. Pascale et al., "Automated testing of healthcare document transformations in the PICASSO interoperability platform," in 2009 31st International Conference on Software Engineering - Companion Volume, May 2009, pp. 163–171. doi: 10.1109/ICSE-COMPANION.2009.5070974.

[12] Z. Lei, Y. Chen, Y. Yang, M. Xia, and Z. Qi, "Bootstrapping Automated Testing for RESTful Web Services," IEEE Transactions on Software Engineering, vol. 49, no. 4, pp. 1561–1579, Apr. 2023, doi: 10.1109/TSE.2022.3182663.

[13] A. Ehsan, M. A. M. E. Abuhaliqa, C. Catal, and D. Mishra, "RESTful API Testing Methodologies: Rationale, Challenges, and Solution Directions," Applied Sciences, vol. 12, no. 9, Art. no. 9, Jan. 2022, doi: 10.3390/app12094369.

[14] S. Ahmed and A. Hamdy, "Automated Black-Box Testing of RESTful APIs Using Enhanced Artificial Bee Colony," in 2023 International Conference on Advanced Enterprise Information System (AEIS), Dec. 2023, pp. 131–135. doi: 10.1109/AEIS61544.2023.00028.

[15] S. Segura, J. A. Parejo, J. Troya, and A. Ruiz-Cortés, "Metamorphic Testing of RESTful Web APIs," IEEE Transactions on Software Engineering, vol. 44, no. 11, pp. 1083–1099, Nov. 2018, doi: 10.1109/TSE.2017.2764464.

[16] D. E. Vega, I. Schieferdecker, and G. Din, "Design of a Test Framework for Automated Interoperability Testing of Healthcare Information Systems," in 2010 Second International Conference on eHealth, Telemedicine, and Social Medicine, Feb. 2010, pp. 134–140. doi: 10.1109/eTELEMED.2010.26.

[17] Stanford Medicine, "PATIENT HEALTH QUESTIONNAIRE (PHQ-9)," 2005. Available: https://med.stanford.edu/fastlab/research/imapp/msrs/_jcr_content/main/accordion/accordion_content3/download_256324296/file.res/PHQ9%20id%20date%2008.03.pdf

Appendix A.

Sample portion of a healthcare application ecosystem

