NIYAM IT™

xformHIT™

Enabling Innovative Patient-Centric Solutions

Transform Health IT

Health IT infrastructure is a specialized system of processing, storing and exchange of critical data, in highly secure and diverse environments, governed by multiple privacy laws and requiring adherence to various evolving interoperability standards. The development of health IT systems requires an approach that is different from other IT systems that makes them sustainable, reliable and future ready to adapt both ways, changing compliance/privacy laws and evolving technology.

Niyam's methodology for designing health information systems, **xformHIT™** is crafted with a laser focused objective of providing open, secure, enterprise-grade solutions. Our extensive experience in high volume data management and analytics, infrastructure and cloud engineering, agile/DevSecOps and GIS applications has been used to distill and formalize **xformHIT™** methodology.

xformHIT™ is composed of 5 key areas:

1. Interface & Interoperability
2. Scalability
3. Security
4. Future readiness
5. Communications and collaboration

The integration of information technology (IT) with health sciences/ healthcare has led to new frontiers and higher benchmarks in operational efficiency. With the adoption of Electronic Health Records (EHRs,) medical imaging, lab data, data generated by Internet of Things (IoT) devices, healthcare organizations both in public and private sector, have never been as keen to incorporate new IT systems that improve their performance.

1. Interface & Interoperability

The fragmented and diverse infrastructure prevalent among most healthcare/health sciences agencies, impairs the ability to work across organizational boundaries.

Measure Interface and Interoperability

We assess the interface and interoperability requirements in stage 0 to build solutions that meet foundational, structural, and semantic levels of interoperability. The interface components are based on RESTful architectures to enable them to exchange information seamlessly and comply with Health Level 7 (HL7) FHIR (Fast Healthcare Interoperability Resources) standards framework.

The categories used for measurement of interface and interoperability:

1. Basic Transactions: The ability of two or more technologies or systems to exchange information. Transaction types to investigate are send, receive, query/ find, use/integrate.
2. Standards used in transactions i.e., Direct, HL7 V2, V3, FHIR, DICOM, GS1

Design RESTful APIs

Based on the interface and interoperability measures, establish requirement definition for the RESTful API model using following steps:

1. Describe the RADM (Restful API Design Model) that enables a model-driven development approach for the creation of consumable APIs. These APIs have consistent resource names, common query parameters, simple JSON/XML data structures that are not deeply nested and do not use polymorphism.
2. Describe detail technical structure of the model and how it relates to other SOA models
3. Develop APIs based on the RESTful API Development Flow

2. Scalability

Scalability demands of a health IT system can be met by ensuring network scalability and storage scalability.

Network Scalability

A major technology barrier in health IT infrastructure is the health information exchange (HIE). Organizations must support HIE on multiple levels, such as provider-to-provider or provider-to home clinician. Health IT systems should be designed for easy upscaling and downscaling without overspending.

Storage Scalability

The growth of picture archiving and communication systems (PACS) has triggered an increase in storage from gigabytes to terabytes.

We determine size of workloads, business processing requirements and performance requirements by measuring and forecasting workloads to accommodate growth, peak transaction volumes, workload complexity, interactions with external systems and network distances to build a scalable solution to exceed Non-Functional Requirements (NFRs). Metrics use to establish scalability requirements:

Transaction profiles describing the size and formats of data exchange

1. Timing of transactions
2. Volume of transactions
3. The architectural solution must follow these principles:

Decomposition of the system: Decompose the system into smaller manageable subsystems, with each subsystem carrying out independent function. The subsystems should be able to run in separate process or threads and enabled to scale using load balancing.

Evaluate asynchronous design: Asynchronous processing enables process execution without blocking on resources but is complex to design and test.

Reduce coupling: Loose coupling provides greater flexibility to independently choose optimized strategies for performance and scalability for different subsystems.

High Cohesion: Reduce the number of interactions between subsystems by logically grouping the classes or components to reside in the same tier. This reduces the need to mix local and remote calls to complete a logical operation.

Parsimony: The solution design must be architected to use system resources as effectively and efficiently as possible by caching, pooling and multiplexing techniques.

3. Security

As responsible and respectful stewards of the data, we address end-to-end security and regulatory compliances by designing systems for audit and access control, centralized identity management, PHI (Protected Health information) encryption and data transmission security. To prevent unauthorized access to PHI we evaluate the possibility of partitioning off all sensitive data and grant access on a user-by-user basis.

Our steps to be used for security technology identification and selection:

1. Identify policies driven by state or federal laws, regulations, organizational policies, business needs or policies developed for specific HIEs.
2. Identify enabling services required to implement the policies.
3. Identify enabling processes that define the operational baseline via use cases and scenarios for enabling services.
4. Design notional security architecture that defines the technical constructs and their relationships to implement enabling processes.
5. Define technical solutions and data standards needed to implement the notional security architecture

4. Future readiness

With change as the only constant, solution design and implementation should align towards increasing levels of connectivity, lower transaction costs and automation. **xformHIT™** ensures continuous improvement and incorporate future enhancements seamlessly.

Our steps to develop future ready solutions are:

1. Formation of a balanced team with team members equipped to identify and prioritize high performance enhancing areas, determine ROI on change implementations and orchestrate rapid implementations for continuous improvements.

2. Maintaining high stakeholder participation by identifying key participants, their style (reactive/proactive) and preferred communication channels.
3. Following solution design principles that include:
    a. Design a light-weight solution to minimize impact of change
    b. Design a simple and intuitive solution
    c. Design for a purpose and stakeholder value
    d. For latest technology stacks and frameworks, design with team learning curves in perspective
    e. Favor open-source software and open architecture
6. Establishing processes to test vulnerabilities, certifying components for reuse and maintaining code quality
7. Evaluating organizational need and feedback after first deployment cycle to assess needs for changes to code, environment, delivery process, automation, and scalability.

5. Communications and Collaboration

Health care is a complex landscape with numerous influencers and drivers. These include governing bodies, compliances, health insurance sector, pharmaceutical industry, research and test laboratories, medical equipment suppliers, clinicians, and administrators. Failure to consider all relevant stakeholders when designing systems results in low adoption rates.

Diplomatic collaboration various professional groups of clinicians, administrators, and IT professionals is essential to implement solutions that instill trust and confidence.

Constructive collaboration among can be achieved by:

1. Focusing on solutions ideas that are effective and feasible to both providers and patients.
2. Discover possible design constraints, by adopting different interview strategies
3. Ensure applicability and relevance by testing conclusions drawn from small user groups on broader populations.

Authors and Contributors

Niyam Health IT Contributor

Rohit Bollineni rbollineni@niyamit.com

References:

RESTful API Design Methodology. (n.d.). www.ibm.com. Retrieved October 18, 2021, from https://www.ibm.com/docs/en/ips/8.8?topic=guide-restful-api-design-methodology

Scholl, M., Stine, K., Lin, K., & Steinberg, D. (n.d.). Security Architecture Design Process for Health Information Exchanges (HIEs). https://nvlpubs.nist.gov/nistpubs/Legacy/IR/nistir7497.pdf

IT Solutions with Integrity, Competency and Responsiveness Responsive Methodology© Making sense of Agile Projects. (n.d.). Retrieved October 18, 2021, from https://niyamit.com/assets/responsive-methodology.pdf

Why Niyam?

A secure, scalable, data-driven IT ecosystem is instrumental in accelerating research, improving operational efficiency, reducing costs and elevating the quality of life. Team Niyam's health IT practice is dedicated to build such ecosystem (or parts of it) that enables our customers in achieving high levels of excellence and a future ready infrastructure that provide a solid foundation for automation, machine learning (ML,) artificial intelligence (AI) and other emerging technologies.

Our Technology Partners

Learn More

For more information on this topic, or to learn about our full range of capabilities, contact us at **703-429-2450** or email **info@niyamit.com**