The Journal of Molecular Diagnostics, Vol. ■, No. ■, ■ 2022

ELSEVIER

the Journal of Molecular Diagnostics
jmdjournal.org

A Model for Design and Implementation of a Laboratory Information-Management System Specific for Molecular Pathology Laboratory Operations

Eban Tomlinson,* Jennifer Goodman,† Margaret Loftus,† Stephen Bitto,† Erica Carpenter,† Richard Oddo,† LuAnn Judis,† Shabab Ali,* Wyatt E. Robinson,* Miranda Carver,† Mariana Ganea,† Kristen McDonnell,† Diane O'Neill,† Jennifer Starbuck,† Eric Johnson,† Erik Meister,† Jonathan Pohl,† Jessica Spildener,† Sheila Shurtleff,† Sheryl Sovie,† Cathleen Melendez,* Pamela Krebs,† Jacquelyn D. Riley,† Christine Wensel,† Caroline Astbury,† Elizabeth M. Azzato,† David S. Bosler,† Jay E. Brock,† James R. Cook,† Yu-Wei Cheng,† Zheng J. Tu,† Michael Cruise,† Walter H. Henricks,† and Daniel H. Farkas†

From Semaphore Solutions, Victoria, British Columbia, Canada; and the Robert J. Tomsich Pathology & Laboratory Medicine Institute,† Cleveland Clinic, Cleveland, Ohio*

Accepted for publication January 18, 2022.

Address correspondence to Daniel H. Farkas, Robert J. Tomsich Pathology & Laboratory Medicine Institute, Cleveland Clinic, 9500 Euclid Ave., LL2-137, Cleveland, OH 44195. E-mail: farkasd2@ccf.org.

The Molecular Pathology Section, Cleveland Clinic (Cleveland, OH), has undergone enhancement of its testing portfolio and processes. An electronic- and paper-based data-management system was replaced with a commercially available laboratory information-management system (LIMS) software application, a separate bioinformatics platform, customized test-interpretation applications, a dedicated sample-accessioning service, and a results-releasing software application. The customized LIMS solution manages complex workflows, large-scale data packets, and process automation. A customized approach was required because, in a survey of commercially available off-the-shelf software products, none met the diverse and complex needs of this molecular diagnostics service. The project utilized the expertise of clinical laboratorians, pathologists, genetics counselors, bioinformaticians, and systems analysts in partnering with software-engineering consultants to design and implement a solution. Concurrently, Agile software-building best practices were formulated, which may be emulated for scalable and cost-effective laboratory-authored software. *(J Mol Diagn 2022, ■: 1–12; https://doi.org/10.1016/j.jmoldx.2022.01.002)*

Data-management needs in clinical molecular pathology laboratories differ in substantive ways from those in other clinical laboratories and anatomic pathology.[1–4] Conventional laboratory-information systems (LISs) historically have not supported, by themselves, the needs of molecular pathology laboratories to the extent that they have in other laboratory disciplines and operations.[2] Molecular pathology laboratories have often relied on a combination of manual methods, spreadsheets, and nonintegrated and/or modular software to meet data-management and operational needs.

Such was the situation in the Molecular Pathology Section, Pathology & Laboratory Medicine Institute, Cleveland Clinic (Cleveland, OH), in early 2017. A revitalization and growth plan for the Section, which included expansion of personnel, equipment, testing platforms, and test development, was undertaken. An improvement deemed fundamental to this re-invention process was a new laboratory information-management system (LIMS) to reduce and eventually replace the outdated, largely paper- and electronic spreadsheet–based data- and workflow-management system.

While the overhaul of the laboratory service was substantial, the focus of this report is limited to a description of

Supported by the Robert J. Tomsich Pathology & Laboratory Medicine Institute, Cleveland Clinic.

Disclosures: E.T., S.A., W.E.R., and C.M. are full-time employees of Semaphore Solutions, the vendor retained by Cleveland Clinic in the development and launch of the laboratory information-management system described in this article.

https://doi.org/10.1016/j.jmoldx.2022.01.002

a replicable process for the modernization of data and workflow management specific to a clinical molecular pathology laboratory. The process employed a customer−vendor relationship. Of necessity, the relationship was a partnership due to the complementary and nontechnical skill sets of each party. The overall goal was to digitize a system based on paper and Excel (Microsoft, Redmond, WA) across a complex, multidisciplinary molecular pathology/cytogenomics clinical laboratory service. This digitization was accomplished through a custom-architected software solution.

This article describes a model for the design and implementation of a LIMS that meets the diverse information-management needs of a full-service clinical molecular pathology laboratory, emphasizes the integral importance of a well-structured development process, and describes a novel application of modern software-based project-management methods and third-party partnerships for building and deploying a LIMS suitable for a modern molecular pathology laboratory.

Materials and Methods

Scope Definition

The goal of the project was to deploy a LIMS to modernize data and workflow management of the clinical molecular pathology laboratory, including cytogenomics. The project encompassed a complete overhaul of paper- and electronic spreadsheet−based data-handling methods into a comprehensive, integrated electronic platform.

Objectives were identified to define the scope and focus priorities of the project:

- Design and implement an electronic information system to support the specialized requirements and workflow of a full-service clinical molecular pathology laboratory, and to fulfill needs unmet by conventional LISs.
- Support the range of clinical testing performed: adult, pediatric, and somatic mutation detection in tumor tissues, blood, and bone marrow; chromosome analysis; germline genetic testing for diagnosis and/or carrier screening; and pharmacogenomics.
- Support the range of complex data management of analytical methods (and applications) in use: PCR; RT-PCR; next-generation sequencing; B- and T-cell gene rearrangement (PCR + capillary electrophoresis); chromosomal microarray analysis; fluorescence *in situ* hybridization; cytogenetic testing; and (genotyping by) mass spectrometry.
- Integrate into the information-technology environment of the institution for accessioning and resulting, particularly the electronic health records system and the conventional clinical laboratory and anatomic pathology information systems [specifically, Sunquest software version 7.2 (Sunquest Information Systems, Tucson, AZ) and CoPath software version 2014 (Cerner, N. Kansas City, MO), at the time, but now in the process of being replaced by Epic Beaker software version November 2020 (Voicebrook, Verona, WI)].

Needs Assessment and Partner Selection

Given the complexity and multifaceted nature of the project, two key needs were recognized: professional project management guided by domain expertise in molecular-testing laboratories, and experienced and expert software-development professionals who could bring to bear state-of-the-art software-development tools and techniques. Between two and four software developers were dedicated to the project at various stages. One business/quality-assurance analyst was included who served as the conduit for the translation of laboratory-specific requirements into software requirements. A Ph.D.-level molecular biologist with project-management expertise was hired to apply process rigor and organization to direct the LIMS development and implementation (and other projects).[5] Laboratory-based personnel participated in their various subject-matter domains as their primary responsibilities to patient care allowed. Partnership with a team of software engineers was established. This team worked with molecular pathologists and laboratory personnel (subject-matter experts) to design and build the full software solution. Peer-to-peer relationships were formed between the clinical laboratory project manager and software project managers as control points for the project.

Prior to the beginning of this extensive development project, due diligence was undertaken to evaluate whether, or to what extent, commercially available information systems could meet the needs at hand. This assessment indicated that some systems on the market could provide some of the functional requirements, and that development entirely from scratch was not required. Next-generation sequencing−centric LIMS software was selected for licensure (BaseSpace Clarity LIMS; Illumina, San Diego, CA). This software as licensed was focused narrowly on supporting particular elements of next-generation sequencing workflows. Its distinguishing characteristics were customizability and extensibility that could ultimately support the objectives of the laboratory, including a wide portfolio of molecular diagnostics tests performed and the need for systems integration. Compatibility with the third-party commercial bioinformatics software (Clinical Genomics Workspace version 6.15.1; PierianDx, Creve Couer, MO) that the laboratory had previously chosen was another key deciding criterion.

Technology Platforms and Software Tools

The software-engineering team used Jira and Confluence web-based software (Atlassian, San Francisco, CA) to organize development tasks and to project documentation,

Table 1 Definitions of Terms for the Agile-Scrum Software-Development Method Applied

Term	Definition
Stakeholders	The group of individuals who work with, and would be impacted by, the software system being developed (ie, laboratory and medical directors, laboratory manager, laboratory supervisors, medical/laboratory technologists, pathologists, geneticists, bioinformaticians, staff scientists, clinical systems analysts, and a revolving cast from the larger Cleveland Clinic information-technology group).
Agile (Jira software; Atlassian, San Francisco, CA) project management	A system of practice to manage project delivery using an iterative approach. Optimization is achieved via continuous releases that include changes based on stakeholder review at each iteration. The process is useful for addressing highly complex problems in a mechanistic, incremental way.
Scrum	Agile framework that encourages cross-functional team progress through short, measured iterations (*https://www.scrum.org/resources*, last accessed November 4, 2021). Each problem is addressed in focused iterations called sprints, in which engineering management and build practices are used for addressing the complexity at hand. Outcomes are predicted and control of risk is assessed incrementally and via empirical observation.
LIMS workflow	Clinical laboratory test workflows have three components: i) preanalytical, ii) analytical, and iii) postanalytical. A LIMS, or other supporting software, digitally models test workflows, storing and classifying large volumes of laboratory workflow data, and automates laborious, repetitive workflow tasks that risk compounding human error. The term LIMS workflows alludes to customized, digital workflow models that span the preanalytical, analytical, and postanalytical stages.
External program plug-in (EPP)	A Clarity LIMS (Illumina, San Diego, CA)–specific term that refers to a standalone script file accessible within the LIMS to perform calculations, transformations, or integrations too complex or cumbersome to configure within the LIMS itself.
Application programming interface (API)	Software intermediary that serves to connect multiple applications allowing them to exchange information. An API dictates what information can be sent and received by a given application and may add security restrictions. It also abstracts underlying code when interacting with other software. An analogy is a person (application A) ordering at a restaurant (application B); the menu represents the API.
Continuous integration (CI) pipeline	The practice of automating the grouping of changes (typically software code but also software-build pipelines and automated tests) from multiple contributors into a single software project; this is software industry best practice, allowing developers to incorporate code changes into a central repository where builds and tests can be run more frequently and easily.
Environments	
Development	System in which new features are actively developed.
Test	System in which newly developed features are tested; user-acceptance testing occurs in this environment; if the test fails, the software goes back to development; if it passes, it is promoted to staging.
Staging	System used for validating changes prior to promotion to production; the system should mirror production in all ways except those new changes to be tested.
Production	The active system processing patient samples; processes PHI and needs to be treated in accordance with HIPAA regulations.

respectively. Version control was implemented in the GitHub web-based software host (San Francisco, CA). A continuous-integration pipeline (Table 1) was built using TeamCity software version 2019.1 (JetBrains, Prague, Czech Republic). PyCharm software version 2019.1 (JetBrains), Docker Desktop Community software version 2.1.0.5 (Docker, Palo Alto, CA), VirtualBox software version 6.1 (Oracle, Austin, TX), and Postman software version 8.1 (Postman, San Francisco, CA) were used to facilitate local testing and development.

Python (Python Software Foundation, python.org), a popular, general-purpose, high-level programming language with well-documented, well-supported engineering standards, was used to develop external program plug-ins, automated test scripts, report templates, and other services. A report-generation and sign-out application named AVRO (analytically validated reporting object; see subsequent sections) was built on a Python server with an Angular JavaScript front-end. For LIMS workflows, the open-source Python s4-Clarity library (*https://github.com/SemaphoreSolutions/s4-clarity-lib*, last accessed August 4, 2021) was used to support batch-analyte data (spreadsheet-formatted) parsing, laboratory-instrument integrations, complex library pooling, de-multiplexing, and other computations on analytes. The Jinja web-based template engine library (*github.com/pallets/jinja*) was used for default report-content templating to support clinical interpretation and clinical-report generation in AVRO.

Project Management

The general project-management process used was Agile-Scrum in Jira, chosen by the owner of the critical path for project completion (ie, the software-development team). In the project, Agile-Scrum was supplemented with additional roles and processes tailored to work in the hybrid clinical laboratory service/software-development consultant environment. It became obvious that definitions and vocabulary

FLA 5.6.0 DTD ■ JMDI1179_proof ■ 5 February 2022 ■ 6:44 pm ■ EO: JMDI-D-21-

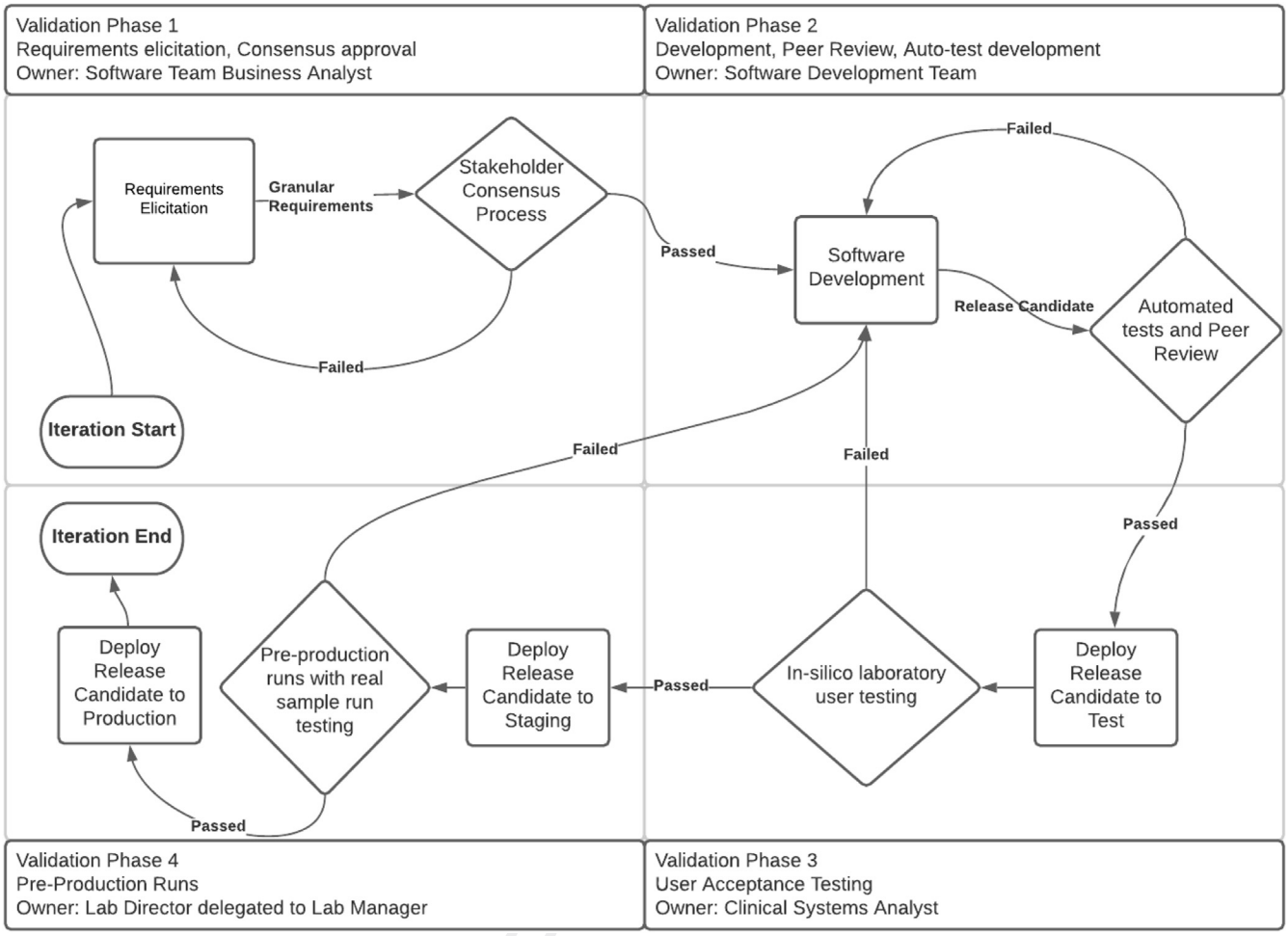

Figure 1 Representation of the four major phases of validation and development.

differed between these two groups of subject-matter experts (ie, clinical laboratory and software-engineering professionals). Selected terms are defined in Table 1. Key roles and activities of the Agile-Scrum development process, as applied in this project, are described below.

Key Roles

Four key roles corresponding to major development and validation phases were identified as gatekeepers of the development process. These gatekeepers controlled the promotion of a deliverable through the process and, thus, control of a rapidly evolving process was retained. The subject-matter experts in these roles worked in tight collaboration to ensure that the highest possible quality was achieved. Note that at relevant steps, bioinformatics professionals, Ph.D.-level laboratory director-designee, and/or M.D.-level pathologists reviewed and approved changes as appropriate.

Key Role 1—Business Analyst (Extant within Software-Development Team)

The business analyst assumed responsibility and accountability for the elicitation and decomposition of requirements into granular requirements that were logically defined and complete. This work output was added to the stakeholder-consensus process.

Key Role 2—Software-Development Team

The software-development team was the consumer of approved granular requirements and the developer of software designed to meet those requirements. This team was involved in developing automated tests to validate that the software written did in fact meet the granular requirements as specified. This work output was collected into a release candidate and, upon passing the automated tests, was released into phase 3, user-acceptance testing. This team was self-organizing and chose the Scrum framework to align themselves into 2-week iterations (ie, sprints).

Key Role 3—Clinical Systems Analyst (Extant within Laboratory Team)

The clinical systems analyst served as the first round of testing for the laboratory service, owing to the proximity of the role to the laboratory and to the expertise in the laboratory's process and informatics architecture. The work assigned to this role was to coordinate and, in some cases, to perform the tasks in development/validation (phase 3), user-acceptance testing. The output of this work was a pass/fail

FLA 5.6.0 DTD ■ JMDI1179_proof ■ 5 February 2022 ■ 6:44 pm ■ EO: JMDI-D-21-

Table 2 Test-Component Organization

LIMS component	Example	Scope
Step	DNA extraction	Single wet-laboratory process contained in a single work session, in one physical location
Protocol	Sample extraction	Collection of steps that facilitate one stage of the workflow
Workflow	Fragile X carrier screening	End-to-end sample processing from accessioning to report issuance

decision that either sent the release candidate back to development or promoted the release candidate to staging.

Key Role 4—Laboratory Manager

The laboratory manager signed documentation approving the release candidate for promotion to the production environment after review of successful preproduction runs that were performed using test samples in the laboratory and after verification that all test plans and checklists were completed.

Key Activities

The major phases of software development and validation [F1] are shown in Figure 1. All activities were coordinated utilizing a series of modular processes that worked together to produce a performant and validated software installation consisting of licensed software, configuration of that software, and customized software to supplement the licensed software [eg, AVRO, Health Level Seven International version 2.3 (HL7; Health Level Seven International, Ann Arbor, MI) interface]. Q35

The highest-order process consisted of four development/validation phases:

Phase 1: Identify requirements and group them into architecturally independent release candidates logically small enough to enable rapid development (requirements elicitation).

Phase 2: Monitor and facilitate the development of automated testing and peer-review process on release candidates (software development).

Phase 3: Coordinate the promotion of release candidates from the development environment to the testing environment (user-acceptance testing).

Phase 4: Coordinate the promotion of the release candidate from the testing environment to the staging environment for preproduction run, and subsequently to the Q36 production *in silico* environments in such a way as to avoid patient-care impacts (preproduction runs, progression to user-accepted version, and deployment).

Development/Validation Phase 1—Requirements Elicitation

The project consumed hundreds of person-hours in elicitation meetings, the goals of which were to:

i. Establish the location of relevant domain knowledge
ii. Establish detailed requirements and shared terminology
iii. Achieve stakeholder group consensus on each granular requirement
iv. Promote granular requirements to the approved backlog of work

Translation of end-user system functionality into workable development items was a task shared between the laboratory and software teams. The laboratory team solicited input from subject-matter experts and prepared documentation. The software business analyst then used that documentation to produce software-development work items that fulfilled the final system requirements specified by the laboratory team. System-requirements documentation included Q37 written requirements, text-based documents, logic tables (eg, user-role permission maps, diagrams, and flowcharts), spreadsheets, native-instrument data files, laboratory standard operating procedures, and test-validation plans.

Software deliverables were detailed and tracked in Jira. Laboratory objectives were described in common language, then passed to the software team for decomposition. Software-development tasks were identified during decomposition and completed by the software team using Scrum. The percentage of completion reports for each objective were tracked using Jira and reviewed weekly by the joint project-management group. Impediments to progress were identified and delegated to the appropriate team members for resolution. Common impediments included:

i. Language ambiguity of requirements
ii. Technical difficulty in mapping laboratory requirements to the software
iii. Requirements without a test plan or for which generating a test plan to be used in practice was challenging

Development/Validation Phase 2—Software Development

Software development proceeded in 2-week iteration cycles (sprints). Members of the clinical laboratory and software-engineering teams continuously evaluated and improved the software-development processes throughout the project. The teams incorporated the requirements of the Health Insurance Portability and Accountability Act (HIPAA) to ensure that protected health information was handled appropriately. For example, scripting of mock samples was developed with convincing, yet mock, data. This scripting enabled the engineering team to test the system in a realistic fashion without the use of patient information.

The engineering team adhered to the following technical process while working within sprints:

i. A team member chose a development task from the approved requirements to work on during the current iteration (sprint backlog).
ii. Once completed, each development task was peer-reviewed by other software engineers prior to

FLA 5.6.0 DTD ■ JMDI1179_proof ■ 5 February 2022 ■ 6:44 pm ■ EO: JMDI-D-21-

Table 3 Shared Workflow Components

LIMS workflow	Workflow components
Cytogenetics (CytoG)	Preanalytics* > (CytoG) wet-laboratory sample extraction > wet-laboratory analysis > clinical reporting > long-term data archiving*
Chromosomal microarray (CMA)	Preanalytics* > (CMA) wet-laboratory sample preparation > wet-laboratory analysis > clinical reporting > long-term data archiving*
Carrier screening	Preanalytics* > sample-preparation extraction* > (carrier screening) wet-laboratory analysis (per test code) > clinical reporting > MDx long-term data archiving*
MDx	Preanalytics* > sample-preparation extraction* > wet-laboratory analysis (per test code) > clinical reporting > MDx long-term data archiving*

*Universally shared component. Note that preanalytics was able to be shared across all workflows, while sample-preparation extraction was shared, but only within the MDx group of tests.

MDx, molecular diagnostics.

inclusion in the master codebase, an industry-standard quality-assurance practice.

iii. Automated tests were generated during development and run before and after each development task was committed to the master codebase, to identify problems in the task and in the integration of the task into the codebase, respectively. Automated tests generally consisted of scripts that generated representative mock samples, and utilized the Clarity LIMS application programming interface to move those mock samples through the workflow while replicating user interaction at each step, again using the application programming interface.

iv. When the development of a full software deliverable was completed (release candidate), it was assembled atop all previous deliverables on a replica LIMS (test server).

v. The release candidate was tested for both successful and potentially erroneous scenarios, in accordance with the requirement specification(s).

Development/Validation Phase 3—User-Acceptance Testing

In phase 3 of software validation, functional testing was performed by the laboratory team and led by the clinical systems analyst, who reported results and discrepancies back to the software team at each stage for collaborative triage. A go/no-go decision was made based on the severity of the discrepancy and the viability of a workaround being utilized until the next release candidate could be deployed. Discrepancies were addressed by the software team in order of priority and redeployed through reiteration of the above process.

Development/Validation Phase 4—Preproduction Runs, Progression to User-Accepted Version, and Deployment

The following process was used for preproduction runs, progression to user-accepted version, and deployment:

i. After all tests were passed, the software vendor deployed the validated release candidate to the *staging environment* of the laboratory.

ii. Successful execution of the user-acceptance testing phase triggered full-system—level preproduction runs by laboratory personnel in the staging environment.

iii. Successful preproduction run completion was followed by scheduled deployment of the release candidate to the *production environment*, which, in some cases, constituted several releases at once.

iv. Full deployment was facilitated jointly by software and laboratory information-technology personnel. Full-system—level tests, as well as preproduction run checklists and end-to-end testing scripts, were utilized to validate the performant nature of the final deployment(s).

Results

System Design—Workflow Analysis and Support

The LIMS organizes and handles the workflows of the clinical laboratory, from specimen accessioning through laboratory analysis to issuing reports to ordering physicians. The system also coordinates tasks among staff. The LIMS automates several laboratory processes, such as spreadsheet parsing or container-placement indexing, and is compatible with laboratory instrumentation, such as automated liquid handlers and quality-control instruments, either through an off-the-shelf or custom-built integration. Substantial customized LIMS-workflow design and engineering were required to model the testing procedures of the laboratory and to organize the constituent processes, including sample accessioning, nucleic acid (DNA, RNA, or total nucleic acid) purification, direct PCR testing, genotyping by mass spectrometry, sequencing, chromosome testing, data analysis, and all associated quality-control steps.

A key project requirement was organization and refinement of workflows within the LIMS. Wherever possible, shared processes were identified and excluded from test-specific workflow building. For example, sample accessioning and DNA extraction were built out as shared protocols. Templating (sharing) at the step and protocol layers

FLA 5.6.0 DTD ■ JMDI1179_proof ■ 5 February 2022 ■ 6:44 pm ■ EO: JMDI-D-21-

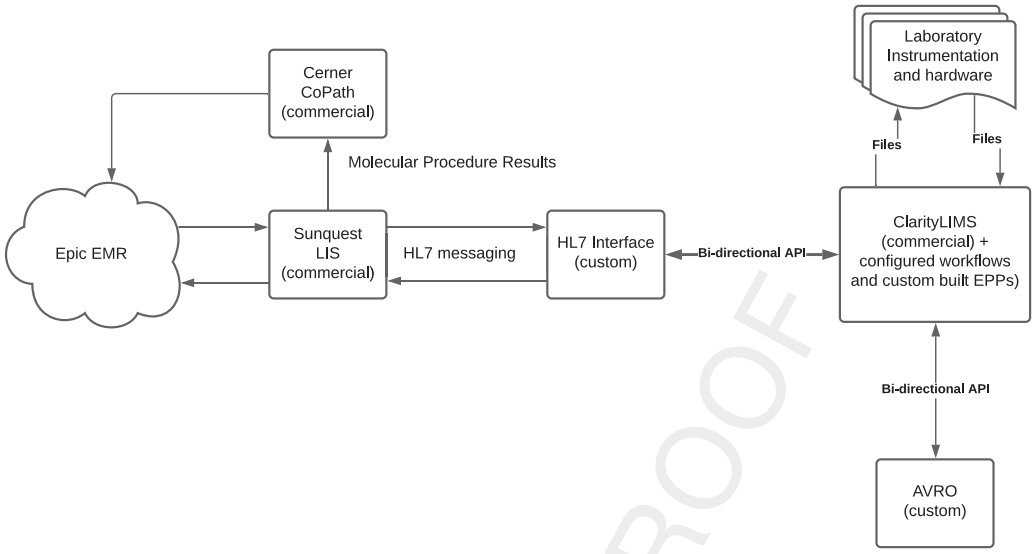

Figure 2 A laboratory information-management system (LIMS) supports clinical laboratory–test workflows systems interfacing and traceability by classifying and storing laboratory workflow data [accessioning, preanalytics, analytical wet bench, analytical interpretation paired to clinical interpretation (results reporting)]. Traceability refers to the record-keeping in each of the systems shown above (ie, data received, data sent, and data transformations that happen in between). AVRO, analytically validated reporting object; EPP, external program plug-in; LIS, laboratory-information system. CoPath, Cerner (N. Kansas City, MO); Epic Beaker version November 2020, Voicebrook (Verona, WI); HL7, Health Level Seven International version 2.3, Health Level Seven International (Ann Arbor, MI); Sunquest software version 7.2, Sunquest Information Systems (Tucson, AZ).

were combined with unique elements to generate end-to-end workflows specific to each test.

The LIMS was engineered to support the build of workflows in a hierarchical fashion in which workflows contain protocols which, in turn, contain steps (Table 2). In steps, the work of the end-user is performed, on either a single sample or a batch of samples. Protocols are a collection of steps organized around the standard operating procedures of a laboratory, and workflows comprise the protocols required to execute a test. Custom-built workflows are shown in Table 3.

System Features and Integration

LIS–HL7 Interface for Sample Accessioning

LIS applications typically connect to other patient health care applications such as Health Insurance Policy Administration systems and electronic health records.[6] To function effectively, LISs must be interoperable with these health care record systems through the adoption of a common standard.[2] The HL7 standard was utilized as the form of communication between the LIS and the LIMS through a customized interface (Figure 2).

The interface between the existing Sunquest and CoPathPlus (Cerner) LISs of the laboratory and the new LIMS streamlined the flow of patient data received from the health care record systems into sample accessioning. A customized HL7 listener service was built and configured to interface with the LIS. This service verifies whether the HL7 message received is valid and, if so, generates a sample with all required sample-information fields and accessions it directly into the LIMS. Required sample fields including, but not limited to, medical-record number, last name, and collection date are extracted from the HL7 message and mapped to LIMS user-defined fields to preserve vital information for mapping test results to the correct patient downstream. The original inbound HL7 message was preserved on the sample to resolve any ambiguous data mapping in the LIMS. Error logs are generated for review when HL7 messages either cannot be processed or produce errors.

The LIS application was integrated with the HL7 interface while its essential functionality and interoperable links were preserved. After software build, the LISs still collected and organized patient health care data. However, software management of the clinical-test workflows and process automation were improved with LIMS implementation.

Instrumentation

Integration with nucleic acid–purification robotics and spectrophotometers (preanalytical) and molecular diagnostics–related analytical platforms were facilitated through file-transfer systems built on shared network locations. Integrations followed the same process:

i. A plate map was generated and displayed to laboratory personnel at the appropriate LIMS workflow step.
ii. After the requisite samples were plated and the instrument completed its full cycle, an output file was directed to a shared network location available at each LIMS workstation.
iii. Laboratory personnel uploaded the output file to the appropriate (usually next) step in the LIMS workflow.
iv. An external program plug-in parsed data from the output file and saved discrete pieces to user-defined fields on the sample in the LIMS.

FLA 5.6.0 DTD ■ JMDI1179_proof ■ 5 February 2022 ■ 6:44 pm ■ EO: JMDI-D-21-

Table 4 Benefits Accruing to a Clinical Laboratory from a LIMS

Item	Category					
	CS	Fin	Ops	PC	Q	Reg
Inventory management and control; optimize reagent purchasing to just-in-time model thereby improving cash flow		✓	✓			
Reduce time needed for archived-specimen retrieval			✓	✓		
Reduce paper consumption		✓	✓			
Monitor process activities: track/locate a specimen as it moves through testing process, waiting times, bottlenecks, instrument usage, assess percentages of instrument capacity being used, idle versus active time, batch sizes			✓			
Reduce opportunities for human error		✓	✓			
Prevent use of expired reagents		✓	✓			✓
Monitor waste, rework, delayed turnaround times	✓		✓			
Optimize use of laboratory human resources; monitor individual clinical laboratory scientist productivity by tracking work units		✓	✓			
Audit trail						✓
Troubleshoot bottlenecks in testing progress; simplification of root-cause analysis		✓	✓		✓	✓
Real-time opportunities for investigation of failures (reagents, instruments, human) and identification of patterns		✓	✓		✓	✓
Capacity to identify rapidly specimens associated with a problematic reagent, instrument, or run			✓	✓		✓
Simplification of monitoring instrument calibration and verification			✓			✓
Capacity to upload supporting documentation so that all data are centralized and not in separate logs			✓			
Monitor volumes by client/physician	✓	✓	✓			
Correlate testing with environment (eg, temperature, humidity)			✓			✓
Correlate technologist training and competence with testing			✓	✓		✓
Centralize instrument maintenance and service agreements		✓	✓			
Opportunity to learn true cost of testing		✓				
Correlate testing done with and without payer preauthorization		✓				
Track test volumes by variables (eg, test code, CPT code, shift)		✓	✓			
Reduce the time that medical technologists invest in performing clerical tasks (eg, pre- and postanalytical clerical tasks, accessioning exceptions, data transcription)			✓			
Simplified, standardized results reporting across all tests	✓		✓			
Reduce time spent on regulatory submissions associated with software		✓	✓		✓	✓
Reduce documentation errors associated with instrument function and verification and workspace decontamination			✓			

CPT, Current Procedural Terminology; CS, customer service; Fin, financial; Ops, operations; PC, patient care; Q, quality; Reg, regulatory.

Clinical Reporting Application

To address the ultimate goal of every high-complexity clinical laboratory test (ie, the issuance of a patient report for the physician who ordered the test), the customized AVRO was built. AVRO serves as an interface to manage the interpretation (postanalytical) component of laboratory-test workflows. AVRO utilizes the authentication system of the LIMS, and therefore only LIMS users with the correct permission setting can access AVRO. AVRO runs on its own server, links directly to the LIMS, and polls specific steps to import reporting objects. It then determines which report template to assign based on the test code and displays test data to professional staff reviewers. This reporting application generates standards-compliant clinical reports for annotation and sign-out by molecular pathology professional staff. Upon sign-out, AVRO sends text reports to the automated clinical report–releasing service, which submits HL7-formatted data and issues a text-based copy of the report to the appropriate recipient.

AVRO was designed and implemented to harmonize results reporting and formatting within a dynamic multicomponent system of EMRs and LISs [Epic Beaker; Sunquest , CoPath, and Rhapsody software version 6.6 (Lyniate, Boston, MA)] while maintaining a flexible LIMS configuration. [Rhapsody is an interface engine that sits between Epic and ancillary systems (eg, Sunquest, CoPath).] Given that report fields are populated directly from LIMS user-defined fields, all LIMS data are available to the report template. Using Jinja2 Python templating enabled a programmatic approach to reporting (ie, rules can be applied to text and data formatting). AVRO uses the LIMS application data layer and the LIMS-managed user credentials that can be integrated with Lightweight Directory Access Protocol systems to support data provenance and added security. This approach obviated the architectural generation of a separate

Table 5 Examples of LIMS Benefits

Item	Benefit(s)	Supporting metric(s)
Specimen demographic information interfaces directly from extant LIS to LIMS-based analytics workflows	1. Preanalytics and analytics technicians no longer need manually transcribe specimen demographic information into Excel (Microsoft, Redmond, WA) logs and workbooks 2. Eliminated time-consuming clerical duties from preanalytics and analytics technicians 3. Eliminated transcription errors that result in discordant specimen demographic information between systems	1. Time-savings, measurements in progress 2. Twenty-four Excel logs used for specimen data storage eliminated and archived
Nucleic acid quantification values and quality metrics parsed directly from NanoDrop (Thermo Fisher Scientific, Waltham, MA) files into LIMS-based analytical workflows	1. Analytics technicians no longer need to manually transcribe quantification data into Excel logs 2. Eliminated time-consuming data transcription duties from analytics technicians 3. Eliminated data transcription errors 4. LIMS-based automations perform quality metrics calculations previously performed by Excel macro	1. Time-savings, measurements in progress 2. One Excel macro eliminated and archived
Nucleic acid extraction reagent and data tracking now completely LIMS-based; previous state required manual completion and archiving of physical worksheets	1. Analytics technicians no longer need to manually transcribe reagent-lot information 2. Reduced paper consumption 3. Reduced physical document storage	Ten high-use nucleic acid extraction documents eliminated and archived
Consolidation of specimen tracking history from multiple systems into one searchable database	1. Higher-resolution specimen tracking enables users to pinpoint specimen location at any step of preanalytical, analytical, and post-analytical processes 2. Expanded and available user-defined fields in extraction workflow reveal whether additional material is available for re-extraction/retesting, ie, computer-based query versus manual search and review in-laboratory 3. Reduced interruption of preanalytics and analytics technicians for specimen-data interrogation 4. Improved troubleshooting and trend identification	Time-savings; measurements in progress
LIMS-based workflow steps include performing technologist, date, and time stamps	Provides high-resolution data previously unavailable to the laboratory which can be used for the identification of technique variation among techs	N/A
Specimen analytical results parsed directly from analytical Excel workbooks into LIMS, which transmits directly to reporting LIS	1. Analytics technicians no longer need to manually transcribe result codes into reporting LIS 2. Reduced data-transcription errors	Time-savings, measurements in progress
Consolidation of reporting system from three separate LISs [CoPath (Cerner, N. Kansas City, MO), Sunquest version 7.2 (Sunquest Information Systems Tucson, AZ), ePath Logic web-based software (ePath Logic, Highland, MI)] to one universal reporting product (AVRO)	1. Standardized reporting process across testing platforms 2. Reduced LIS access and training requirements for analytics technicians	Reporting LIS reduction from three systems to one
LIMS transmits data directly to Tableau software (Tableau, Seattle, WA)	Enables more sophisticated queries and data analysis for quality metrics reporting and troubleshooting	N/A

(*table continues*)

FLA 5.6.0 DTD ■ JMDI1179_proof ■ 5 February 2022 ■ 6:44 pm ■ EO: JMDI-D-21-

Table 5 (*continued*)

Item	Benefit(s)	Supporting metric(s)
Automated calculation for reagent addition in FISH	1. Saves time 2. Eliminates arithmetic errors 3. Automates recording of probe(s) used, lot numbers, expiration dates	Time-savings, measurements in progress
Case tracking (FISH laboratory)	1. TAT tracking automated 2. Rush samples or samples approaching TAT limit more easily sorted and found 3. Facilitates case assignment to technologists	Time-savings, measurements in progress

FISH, fluorescence *in situ* hybridization; TAT, turnaround time.

database and the duplication of user credentials. AVRO thus facilitates professional actions on samples that exist entirely within the LIMS.

The concept of report-content defaults was identified from feedback after early implementations had been tested by the reporting professional staff. Reporting templates were customized for each test and then served the correct template by mapping test codes to those templates. Templates could be mapped to multiple test codes and were maintained independently for maximum flexibility in content.

Opening a report in AVRO presents a template based on the above logic, populated with interpretation text derived from the results-interpretation code applied in the analysis portion of the workflow. The molecular pathologist user is able to modify and customize all of the populated content prior to sign-out. At any point during the report-building process, AVRO allows the user to preview the report.

A customized report-release service was built into the LIMS as a set of external program plug-ins in a report-release step. This step was designed such that the system monitors for newly signed-out reports in AVRO every 30 seconds via the AVRO application programming interface. The report-release step generates and sends reports back to the LIS, and then attaches the report file(s) to the completed step in the LIMS.

Audit trails were available as base functionality in the LIMS and did not require customization. Building AVRO atop the audit trail—data layer facilitated auditing through the report-building process up until submission of the outbound HL7 message. HIPAA compliance as implemented in the LIMS thus persisted through the report-building process, an efficiency afforded by leveraging of the existing software-validation work extant in the off-the-shelf product. The LIMS tracks all of the elements specified by audit trails—related items on the College of American Pathologists Laboratory General Accreditation Checklist.

Resulting End-to-End Software Solution

The test-managing software system of the LIMS and AVRO interfaced with the pre-existing LIS architecture via customized software services, an HL7-bridged laboratory sample—accessioning service, and an automated clinical report—releasing service through the LIMS. The newly integrated solutions automated the full end-to-end process of the molecular pathology laboratory, from sample accessioning to the issuance of final patient reports. These services eliminated the use of paper to track analytics data and mitigated the risks for bottlenecks and human error that tend to result from manual control of laboratory input and output processes. For example, accessioning bottlenecks have been mitigated by streamlining accessioning through the LIS—HL7 interface, and human errors due to data transcription have been reduced through data tracking in the LIMS and subsequent automated exchange of data between the LIMS, AVRO, and LIS.

An important predictor of success was the selection of a LIMS with robust application programming interface and customization support in order to address the significant Δ between off-the-shelf functionality[2] and the functionality required by the molecular service via the only pathway remaining, customized-software engineering. Software engineers worked in virtual instances of the software environment to enable rapid revisions within existing and relevant system parameters. Imagining, designing, testing, and implementing the information-management system described required significant human and financial resources. A list of advantages that have been accrued by this clinical laboratory, or that are anticipated in the near future, is shown in Table 4. Specific examples of the benefits the laboratory has experienced since LIMS implementation are listed in Table 5.

Discussion

The Molecular Pathology Section, Pathology & Laboratory Medicine Institute, Cleveland Clinic, performs high-complexity clinical laboratory testing in support of the practice of genomic medicine. A revitalization and modernization program begun in 2016 established the need for improvements in laboratory data management. Organization of clinical laboratory—testing workflows in the 21st century is best matched to 21st-century software and information-management tools, not tools from the previous century (ie, paper, Excel, flash drives). Education was pivotal in accelerating the project toward success. Educating all members in this team-based approach in respective

domains led to the acknowledgment that both software development and molecular pathology (with inherent regulation) were of equal importance in the LIMS-generation process. Education, not unlike the build process itself, was accomplished iteratively and continually throughout the project. The more that subject-matter experts from all disciplines understood each other, the better the team was able to optimize and improve each build iteration and, subsequently, execute the shared process successfully. The importance of partnership between the laboratory and software-engineering teams was a lesson well learned.

Modeling clinical laboratory–test workflows digitally and integrating a LIMS into the clinical hardware and software architectures of the laboratory required a significant amount of customized-software engineering. Most LIMSs available commercially are touted as configurable. Principles and practices from the domains of computer science and software engineering were required to produce a maintainable and scalable LIMS implementation, alongside a greater set of laboratory software services and applications. Laboratories often do not have the resources in-house to integrate such a solution into their existing system(s). Software and process engineers were required to build effective customized test-workflow models *in silico* that automated processes and managed the inherent workflow data. Their role also included installing and successfully integrating the LIMS application alongside other supporting software that existed in the clinical-informatics architecture. By employing a clinical laboratory team/software-engineering team collaboration, the existing software system was significantly upgraded, the use of paper and electronic spreadsheets was reduced (with the eventual goal of the complete elimination of paper), and efficiency was increased, putting the laboratory in a better position to scale up throughput and manage complex data.

The test-reporting environment is complex; here is an example of one benefit of the LIMS implementation. The Pathology & Laboratory Medicine Institute (composed of two departments, [anatomic] Pathology and Laboratory Medicine) uses Sunquest to result laboratory medicine testing. CoPathPlus is used to report anatomic pathology results and additional associated testing of tissue, cytology, and bone marrow specimens (ie, special stains, fluorescence *in situ* hybridization, cytogenetics, PCR, and next-generation sequencing testing). Clarity LIMS streamlined reporting so that all molecular pathology tests are now reported back to Sunquest and posted as a discrete item in the patient's electronic health record in Epic for the specific test. Tests that must be reported as part of an anatomic pathology report are sent back to CoPath from Sunquest so that clinicians may view a comprehensive report on that specimen.

Formulating an effective software-building framework was important, considering the challenges in generating and operating customized clinical laboratory software. Such software must address testing intricacies and diverse molecular technologies within a complex clinical laboratory architecture. Rigorous building methods and techniques were needed to ensure build quality and maintenance sustainability.

Scrum best practices are valuable in the modernization of clinical laboratory information–management via customized-software generation. Scrum best practices are demonstrated to be extensible, scalable, and cost-effective. It is crucial that any given Scrum format: i) be lightweight, with clearly defined roles and responsibilities of the participants; ii) be comprehensively simple; and iii) assume room for improvement.

Notably, a laboratory-science framework for molecular biology research, termed *LabScrum*, has gained support in research-oriented molecular biology laboratories. LabScrum emulates the principles of software-engineering Scrum. The writers of LabScrum illustrated three critical principles of the Scrum framework's utility in research laboratories.[7] These principles are valid for building customized information-management software in the setting of clinical laboratory testing.

The adoption of some form of Scrum in building customized laboratory software should focus on three essential components: i) transparency, ii) inspection, and iii) adaptation. LabScrum proponents have argued that a management framework based on empiricism is highly consistent with the scientific method.[7] Anecdotally, research scientists rigorously apply empiricism and the scientific method for results, or product, but often do not think empirically about tasks/techniques, or processes. Generating process visibility and analysis utilizing a framework for ongoing improvement can improve the quality of the science (more rigorous science), the quantity of the science (more productive science), and the quality of life of scientists (more sustainable science). In effect, to run a modern clinical laboratory necessitates having a laboratory function to interact with the processes of configuring or generating customized software, because, by definition, this software does not yet exist or, more accurately, must be customized for use in every clinical laboratory.

The inability of the LIMS implementation to store unique patients in their own table with a one-to-many relationship between the patients table and the specimens table in the database is a limitation of the existing system. The system is maintained via Information Technology department personnel and 2.5 in-laboratory medical technologists.

An important lesson learned in executing the LIMS project was the concept of a minimally viable product (MVP). Early in the project, the laboratory team was excited at the flexibility and customizability of the chosen LIMS product. In understanding what was possible and what was needed to elevate the laboratory's capabilities, overreaching, unrealistic goals were imagined. The team grew too ambitious, frustration accompanied scope creep, and project velocity decreased. Once it became clear that an MVP was, by definition, good enough, goals became simultaneously more modest and attainable. Thus the project's progress toward full execution accelerated over the last approximately 12 months

FLA 5.6.0 DTD ■ JMDI1179_proof ■ 5 February 2022 ■ 6:44 pm ■ EO: JMDI-D-21-

of a 30-month effort from late 2017 to early 2020. With LIMS launch, experience, and newly acquired local skills and knowledge, the laboratory team looks forward to independently adding more functionality via new customizations.

Acknowledgments

We thank Gordon Meyer, Justin Chant, Emily Wong, Jeremy Snell, Danny Hopkins, Mark Swinkels, Mark Luszniak, and the rest of the Semaphore team for their technical expertise; Wendy Nedlik for organizational-management expertise, Hillard Meade for project-management expertise; James Fenske for software-implementation assistance; Michael Reese for financial analysis; Eric D. Hsi, MD, for leadership; and Jacqueline Urankar and Susan Brennan for excellent administrative support.

References

1. Kang W, Kadri S, Puranik R, Wurst MN, Patil SA, Mujacic I, Benhamed S, Niu N, Chao JZ, Ameti B, Long BC, Galbo F, Montes D, Iracheta C, Gamboa VL, Lopez D, Yourshaw M, Lawrence CA, Aisner DL, Fitzpatrick C, McNerney ME, Wang YL, Andrade J, Volchenboum SL, Furtado LV, Ritterhouse LL; Segal JPSystem for Informatics in the Molecular Pathology Laboratory: An open-source end-to-end solution for next-generation sequencing clinical data management. J Mol Diagn 2018, 20:522–532
2. Myers C, Swadley M, Carter AB: Laboratory information systems and instrument software lack basic functionality for molecular laboratories. J Mol Diagn 2018, 20:591–599
3. Lee RE, Henricks WH, Sirintrapun SJ: Laboratory information systems in molecular diagnostics: why molecular diagnostics data are different. Adv Anat Pathol 2016, 23:125–133
4. Campbell WS, Carter AB, Cushman-Vokoun AM, Greiner TC, Dash RC, Routbort M, de Baca ME, Campbell JR: A model information management plan for molecular pathology sequence data using standards: from sequencer to electronic health record. J Mol Diagn 2019, 21:408–417
5. Brock JE, Nedlik W, Farkas DH: A process for new clinical laboratory test implementation. J Mol Diagn 2018, 20:1016, Abstract
6. Sinard JH, Gershkovich JH: Custom software development for use in a clinical laboratory. J Pathol Inform 2012, 3:44–53
7. May L, Runyon T: LabScrum: a case study for agility in academic research labs Cutter. Arlington, MA, Arthur D. Little, June 25, 2019. Available at, https://www.cutter.com/article/labscrum-case-study-agility-academic-research-labs-504061 (accessed Month day, year) Q54

FLA 5.6.0 DTD ■ JMDI1179_proof ■ 5 February 2022 ■ 6:44 pm ■ EO: JMDI-D-21-