

Healthcare Interoperability through Rigorous HL7 Testing

A Comprehensive Review of HL7 testing methodologies

Shreekant Majge

Sr Mgr, Quality Assurance: IT-Data Tech Services,
Radiology Partners Inc
USA

Abstract— Interoperability has been always a challenge for healthcare organizations and testing the exchange of data using HL7 is very critical to keep clinical and financial systems accurate. In an era where healthcare data is integral to patient care, HL7 has emerged as a foundation for standardized information exchange. This article highlights the critical role of testing in ensuring the interoperability of healthcare systems, emphasizing the need for a robust HL7 testing. This article also presents a comprehensive review of testing methodologies used in integration of healthcare applications. We discuss the significance of HL7 testing, outline key testing approaches and best practices.

Keywords— HL7, Health Level 7, Testing, Interoperability, Healthcare Information Exchange, Electronic Health Records, Quality assurance.

I. INTRODUCTION

Implementing interfaces to exchange clinical and financial data is critical to any healthcare organizations. Hundreds of government-certified EHR products are in use across the country, each with different clinical terminologies, technical specifications, and functional capabilities. In fact, not even those EHR systems built on the same platform are necessarily interoperable because they are often highly customized to an organization's unique workflow and preferences [1].

When we start the discovery of integrating two systems the source system has the codes which are mapped to standard and the receiving system has to interpret these standards back into their coded values. Lack of standardized data makes it more difficult to share data electronically for patient care. To address these challenges HL7 organization provided a framework (and related standards) for the exchange, integration, sharing, and retrieval of electronic health information. These standards define how information is packaged and communicated from one party to another, setting the language, structure and data types required for seamless integration between systems [2].

Testing the integration between systems is not limited to check the conformance to the HL7 standard, we also need to make sure the seamless integration occurs and the data is well formatted, cured and good quality for patient care, billing and analytics.

II. IDLC: INTERFACE DEVELOPMENT LIFE CYCLE

Before we review the testing methodologies, we need to understand the Interface development life cycle which explains the overall process and handshaking of the teams and processes involved in Implementing the interfaces for healthcare interoperability. The IDLC mainly consist of 5 stages and I will brief on what is involved in each stage and how these are interconnected.

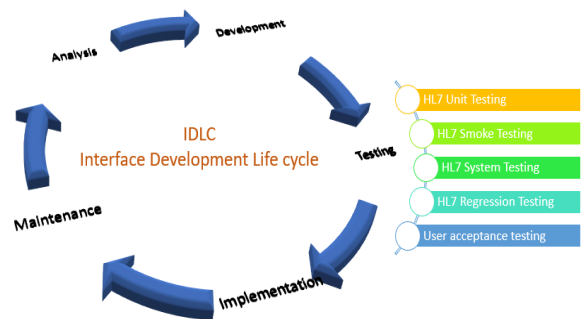


Fig. 1. IDLC – Interface Development life cycle

1. Analysis

This is the first stage of interface development where client reach out to integrate the healthcare systems. Typically, business analysis teams work with client to gather all the interface requirements, application HL7 specifications and HL7 messages to analyze. Business analysis team go thru all the requirements and analyze the messages with all the data patterns and work with client to review all the mappings to build the Integration work book. As soon as Business analyst generate the workbook of requirements to build the interfaces, it will be handed over to HL7 Developers for building the interface.

2. Development

During development phase the HL7 developer review the workbook generated by business analyst and start building the interface using the interface engines which uses the HL7 standard to exchange the data. Developer creates all the logic for mapping, error handling, data transformations, storage etc. and will start the HL7 unit testing.

HL7 Unit Testing: This is the starting phase of validating the working product where Developers make sure the Interface requirements are coded and validated before handing over to QA team to perform functional testing. It is recommended that the peer review panel performs the design and/or code inspections and developer run the HL7 messages through interface to make sure the required logic is built and working as expected. This is the critical step in SDLC where developer needs to verify the logic is built to make sure it handles all the exceptions, negative data checks and HL7 standard. Usually, Developer does not perform the unit testing thoroughly which will lead to delay the schedule and QA team may ignore to test the code exception which may fail the production build which eventually impact patient care. Unit testing should be added as part of the precondition before the build is handed over to QA for further testing.

3. HL7 Testing: Review of Testing methodologies

The success of Interoperability between systems heavily depends on rigorous HL7 Testing and this section delineates the testing methodologies used in HL7 Testing. The systematic testing of integration should occur in systematic testing techniques. The success of HL7 Interface implementation highly depends on the efficacy of testing methodologies employed to validate the implementation precisely.

The following four stages of testing have been efficient and widely used for HL7 implementation projects.

A. HL7 Smoke Testing

A test suite that covers the main functionality of a component or system to determine whether it works properly before planned testing begins [3]. Developer's handover the interface built for testing to QA team. It is always recommended and best practice to perform the HL7 smoke testing. In current rapid development and deployment trend it is key requirement for every organization to run the UAT in parallel to detailed testing. HL7 smoke testing phase QA team run the HL7 messages from source system and make sure the messages are processed successfully and display the data in all the connected downstream. During smoke testing QA will focus on the testing workflow and make sure there are no show stoppers to continue with full system testing along with UAT. This approach has always shown significant reduction in test life cycle time, early feedback and timely delivery. HL7 Smoke test cases should be created carefully to make sure the system interfaces, data flow and all the required fields are mapped. QA team should not spend more than 4-6 Hrs on HL7 smoke testing and always the expectation here is to make sure the Interface is built to test, deploy and run the HL7 messages.

B. HL7 System Testing

A test level that focuses on verifying that a system as a whole meet's specified requirements [3].

QA team now deep dive into testing every aspect of system in this phase. HL7 System testing is divided into several testing areas where QA team test end to end scenarios where the HL7 data flows from source to destination with all the Interface transformation, data mapping, source HL7 fields to destination mapping, Database completeness, application validation with HL7 data etc. During system testing QA team member should generate test HL7 messages or scrub the production messages for PHI and process them through test systems to make sure the workflow is built and working as expected. HL7 system testing need to make sure that the seamless system integration occurs and the data is well formatted, cured with good quality to make sure patient care, billing and analytics are performed.

C. HL7 Regression Testing

A type of change-related testing to detect whether defects have been introduced or uncovered in unchanged areas of the software [3].

HL7 regression testing is crucial in every integration of systems where we use HL7 standards for mapping data fields than the traditional integrations not using HL7. Any type of code or mapping change in interface needs system testing and most importantly hl7 regression testing to uncover the defects in unchanged area or any downstream application interfaces. Some of the QA organizations run regression test at the level of system testing and verify the downstream like traditional application regression tests. There are issues with just running few messages thru the interface and verifying the downstream since the data and mappings can be reviewed only when you load set of data with simulated environment.

Any small change in code also needs to be thoroughly regression tested in very methodical way to achieve the best results and uncover all the defects to avoid any potential defects in production which may impact patient care. The standard and application HL7 mappings are highly dependent and very impactful when it comes to patient care workflows. A simple change in patient class mapping from Emergency to Inpatient may cause significant delay in treatments and care which is needed for the patient.

A comprehensive understanding of HL7 regression testing contributes to the resilience and data integrity of healthcare systems. During this phase the production HL7 messages needs to be processed in staging and the outbound messages of each downstream interfaces needs to be compared for differences as shown in Fig.2. If there are differences in the compare process then those should be reviewed with developer to understand as those are expected differences vs unexpected. The expected difference needs to be ignored like the number 151 in fig.2 is expected and unexpected difference should be reported back to developer like shown in the Fig.2 where the line starts with PV1 has the red highlighted text to review and fix the code. When Developer fixes the code the QA team should re process the messages in staging and compare again until all the differences are expected.

```

MSR|^~\g|Ntierprise|Ntierprise|EHR|Clinic|20190423113910||ADT^A08|8899-;
EVN|A08|20190423113910||01
PID|1|151|Bond^Tiny|19990723|M||8388 Secret Agent Way^Raleigh^NC^2
NK1|1|Bond^Lady^T|Spouse^Spouse|007 Soho Lane^Cary^NC^27511|(919)007-0
PV2|1|R|1|1|Manning^Manning^Terry^6765432140PDM|Manning^Manning^T
GT|1|150|Bond^James^007|007 Soho Lane^Cary^NC^27511|(919)007-0007^*

```

Fig. 2. Comparison of HL7 message Staging vs production

This article advocates for the implementation and best practices used for the advanced testing methodologies to validate and optimize HL7 standards, promoting a secure and efficient electronic data exchange. As the healthcare industry progresses, embracing rigorous HL7 testing practices becomes indispensable for delivering high-quality patient care, billing and operations.

D. HL7 User acceptance Testing

A type of acceptance testing performed to determine if intended users accept the system [3].

When Smoke, system and regression Testing performed by QA team may not have access to client applications to verify the data in all the downstream applications processed and displayed as expected. In this phase Implementation team works with client run the complete clinical and billing workflow to make sure the system behaves as expected and the client sign off on the data, formatting and mapping standardization to proceed with Implementation.

4. Implementation

After the analysis, development and detailed testing is complete and approved by client to move the changes to production then release management team will move all code, database and mapping configurations to production. When Release management team deploys the code then it will be notified to all the stakeholder to perform the checks and confirm the interface is now generally available to start utilizing for the patient care, billing and analytic purposes. As users start using the interfaces the changes move into warranty phase.

5. Maintenance

As soon as warranty is over then the interface will be handed over to support or operations for ongoing maintenance. During this support phase the defect will be triaged and fixed, if there are any change in existing feature or mapping then the IDLC kicks in to follow the cycle from analysis to delivery.

III. FUTURE PROSPECTS AND CONCLUSION

Looking ahead, the article speculates on the automation tools and utilizing the tools to run the HL7 regression testing to quickly uncover the issues and speed to implement the interfaces. Considering emerging technologies integration with artificial intelligence and machine learning algorithms there is a potential impact on healthcare system resilience and adaptability.

REFERENCES

- [1] Reisman M. EHRs: The Challenge of Making Electronic Data Usable and Interoperable. P T. 2017 Sep;42(9):572-575. PMID: 28890644; PMCID: PMC5565131.
- [2] "Health Level Seven International hl7.org".
- [3] ISTQB Standard Glossary of Terms Used in Software Testing" https://glossary.istqb.org/en_US/search

DISCLAIMER

This disclaimer informs readers that the views, thoughts, and opinions expressed in the paper belong solely to the author, and not necessarily to the author's employer, organization, committee or other group or individual.