Background

Many healthcare organizations are migrating from one vendor’s electronic medical record system (EMR) to another vendor’s in an attempt to receive federal funding, improve their organization’s performance and quality of care, and communicate more effectively with community physicians and ancillary care organizations.

A system migration is a huge undertaking. One of the most complicated aspects of a migration is data conversion. Historical patient data must be moved to the new system, yet the new system most likely cannot receive the data in its present format. Therefore, the data must be extracted from the legacy system and transformed into a format that will work in the new system – all while the healthcare organization continues to operate.

Choices regarding which data is moved and how are made on a case-by-case basis. Which data should be moved to the new EMR? Which could be stored in a data warehouse? Should the legacy system be retained in a read-only state? Should we do chart abstraction? Scanning? For which charts? Which data is unusable, unnecessary, or otherwise does not have to be moved?

In this paper, we provide a high-level overview of a conversion project, accompanied by best practices. The majority of the best practices are vendor-neutral. However, we have included some that are specific to EpicCare Ambulatory EMR, as we have most experience in this area. As an Epic preferred partner, we sought and received approval from Epic for this content.

It is important to establish up front that conversion projects are not linear, but iterative. All phases feed new information back to previous phases, which can then trigger changes throughout. This iterative process continues until each part of the conversion has successfully passed the testing stage.

That said, Hayes’ conversion methodology is comprised of five phases as depicted in the graph below: 1) Discovery, 2) Requirements, 3) Build, 4) Test and 5) Go-live. Some of these phases overlap and timelines vary by organization. The following graph depicts these phases.

Figure 1: Data Conversion Phases
Phase I – Discovery

The discovery phase sets the foundation for the entire project. It is important to be extremely thorough here as it will make the rest of the project progress more smoothly.

Discovery activities include:

- Gathering initial data (i.e., workflows, data flows) specific to interfaces and data conversions
- Establishing project governance and project teams
- Determining the high-level scope baseline, project success metrics, roles and responsibilities, change control practices and document management
- Developing an initial project plan and drafts of the risk and communications plans
- Conducting preliminary analysis of existing documentation
- Analyzing the structure and content of all source conversion systems
- Reviewing and analyzing the target data model
- Conducting a validation assessment

**Best Practices: Discovery**

**Assumptions:** Delineate the project’s dependencies such as the build, test environments, system resources for full volume testing, etc.

**Risks:** Make a list of risks and their contingency plans. What would jeopardize your deliverables or schedule? (Some examples include: the build is not available as scheduled, late discovery of data issues, un-mappable data that cannot be loaded, etc.) Verify commitments from outside work groups. Diligence in this area will pay off in the long run.

**Service level agreements:** Agreements need to be in place among different project teams for turning around decisions and resolving issues that affect the conversion effort. We highly recommend that service level agreements be put in place that specify expected timeframes.

**Data content issues:** Plan for late discoveries of data content issues. Many are systemic in nature; others are random, resulting from arbitrary system usage. Good analysis can uncover the majority of these patterns prior to and during conversion design. However, some portion of these can be expected to go undetected until they are revealed in testing. Your plan and timeline needs to anticipate these challenges.

**Governance:** Often referred to as an executive steering committee, this group’s members should be comprised of decision makers from clinical, administrative and IT areas. They will be responsible for championing the conversion and providing the necessary senior leadership support. They will also be the ultimate arbiters of tough decisions. Define a charter to clarify the role of governance, and choose people who are respected and can together carry the message to and from all stakeholders.
Conversion and mapping teams: The project team requires a combination of people with clinical, technical and project management skills. Here are our suggested team roles:

- Clinical SME for the legacy system
- Data extraction SME/developer for the legacy system
- Interface engine SME/analyst(s) and developer(s)
- Conversion architecture SME
- Project management SME
- Issues and change control SME

If you are migrating to EpicCare Ambulatory EMR, you will need EpicCare clinical SME(s) and analyst(s), and EpicCare Bridges SME/analyst(s).

The data mapping team will be a conversion project sub-team. It should be comprised of:

- Data analyst/reporting specialist(s) to assist mappers
- Pharmacist & pharmacy tech. (medications, allergens and immunizations)
- Nurse(s) and/or coder(s) (problem list, procedures, other results, flow sheets, histories, directives, health maintenance, others)
- Radiology coding SME
- Laboratory coding SME
- Site SME(s) (providers, locations, document types, visit types, transcription types, preferred pharmacies, others)

Phase II – Requirements

During this phase, detailed interface and conversion requirements are gathered and approved. All functional and technical requirements are documented. Formal requirement walkthroughs are conducted, and gaps are inventoried and addressed.

Detailed design documents are the basis for walkthroughs; they receive preliminary approval in this phase. The initial test and cutover plans are created and specific processes related to the build and test cycles, defect resolution and testing sign-off are incorporated into the resulting requirements deliverables.

Best Practices: Requirements

Conversion architecture: Always look at the conversion of data from a legacy system in the context of three architectural components and their associated processes: 1) extraction of legacy system data; 2) transformation of data in the interface engine, and 3) insertion of the newly formatted data into the new system. The following diagram illustrates a simplified EpicCare conversion architecture using a big bang implementation approach. In a rollout approach, the design would be expanded substantially to include numerous real-time conversion interfaces that would need to run throughout the rollout period to transfer all manual entries from the legacy system.
Figure 2: Sample EpicCare Conversion Architecture (simplified)

**Specification sequence:** We recommend the following sequence: 1) Work with your new EHR vendor’s interface analysts to determine your interface strategy and tasks; 2) design the legacy data extract, and 3) analyze the interface engine transformation. The primary job of the interface engine in an Epic conversion, for example, is to transform the raw extract data into formatted HL7 messages that meet the detailed specifications appropriate for a transaction to be loaded into Epic by Bridges.

**Data mapping:** Plan to deliver the EHR build based on the schedule needed for the conversion data mapping effort. Schedule the mapping effort based on mapping volume and complexity. Make sure you have the appropriate clinical and content expertise available to accomplish each data mapping.

**Data mapping in the EpicCare technical design:** Data mapping works in unison with these primary components by mapping legacy data to EpicCare values in one of three ways:

1. We recommend using Identity, Epic’s MPI module, as the first choice. It provides the ability to associate multiple Epic code sets with entries in an Epic master file.
2. Our second choice would be mapping tables in Bridges, Epic’s interface module. Bridges tables can be configured so that while a transaction is passed through Bridges, Bridges can look up a value in a cross reference table and deliver the mapped value to the Epic system.
3. As a third choice, you can map tables in the interface engine. The MPI and Bridges tables should be used whenever possible. However, in the rare occasions that neither the MPI nor a Bridges table is available to do the mapping, the table can be built in the interface engine and the cross referencing performed before the transactions reach Bridges.
Design and execution sequence: Your go-live approach must be determined in the discovery phase as it affects design. Clinical design must come before technical design. Implementation design must be complete before conversion design. The build has to be complete to start the mapping process, and certainly before testing.

Documentation: Document everything. There are countless decisions made during a conversion project. It is important to write them all down for easy access and reference. This includes: status reports, dashboards and other visual aids, discovery analysis and recommendations, preliminary and final conversion architecture design, detailed design specifications, data mapping analysis and final data mappings, decisions, design changes, issues and change orders.

Detailed design specifications: It is highly recommended that you obtain input from people in many different areas of the organization so that each individual can contribute in his/her area of expertise. The final results of these individual analysis steps are then combined into a comprehensive set of specifications that become the detailed blueprints for building the conversions.

Format: Many clients have standards for specification formats. Since there will be many specifications, it’s important to enforce a standard so that all specifications are consistent. We recommend that the conversion team review the specification format with stakeholders to assure that all aspects of design are accounted for in the standard before the specification writing begins.

Content: Every detail must be defined on a field-by-field basis. The specifications need to start with describing how the legacy data will be extracted and transformed, and end with a segment-by-segment, field-by-field HL7 specification that meets the new system’s requirements.

For conversions to EpicCare, data must have the ability to be loaded using a Bridges interface. Additionally, Bridges interfaces in Epic can behave differently based on variables referred to as “profile variables.” The specifications must indicate the profile variable settings required in each Bridges interface. It is important to have someone who is certified and experienced in Bridges to assist with profile variable decisions and implementation.

Methodology and participants: The methodology for developing comprehensive specifications starts with an assessment of legacy system data content and determining how that content will reside in the new EMR. This initial assessment requires participants who know their respective systems from a user and a build perspective.

Specification documentation should be updated and distributed after each design session. Each participant should keep up with reviews of the documentation to assure that everyone is on the same page.

The importance of the collaborative nature of the design should not be underestimated. Ideally, this collaboration should include key stakeholders, including those with subject matter expertise on the current use of the legacy system and on the expected usage of the new EMR. Many conversion decisions depend on EHR implementation design decisions; it is during this conversion design stage that input from the new system’s implementation design is critical.
Phase III – Build

The build phase begins as soon as the core architecture and planning tools are firmly established, which is prior to the completion of the requirements phase.

Build phase activities include:

- Building, configuring, and unit testing all interface and data conversion initiatives.
  - Data extracts
  - Data transformation
  - Data loads
- Working with relevant third-party vendors to develop, test and implement components.
- Identifying, analyzing, tracking and reporting system and other related problems and issues as found.
- Maintaining specifications and other documentation when iterative design changes take place.

Best Practices: Build

Mapping prerequisite: A conversion requires early delivery of many components of the new EMR system, and therefore requires an iron clad agreement that the EMR build will be accelerated to meet the needs of the conversion (See Discovery Phase, Service Level Agreements). The build must be in place to allow legacy data to be mapped to the new system’s structures. Without early delivery of the build, the mapping effort will stall the conversion testing effort while the build and mapping efforts catch up with the technical conversion effort.

Mapping: There will be multiple versions of mapping documentation. It is important to manage these so that team members always have the latest version available to them.

Mapping is never static. During the course of the conversion, new values will be used in the legacy system and the mapping effort needs to include mapping iterations to identify the new values and make the additions to the mapping tables.

Some mapping is quite time consuming. For example, free text is not a primary option in Epic. We’ve encountered instances where spelling, minor grammatical differences and errors have accounted for 90% of the mapping effort.

Conversion design: In-depth analysis should precede design. It is important to know who will be using the new system. Analysis of the legacy EMR needs to be performed to understand the style and nature of the data. The conversion design needs to be based on the clinical record design coupled with a technical design that meets clinical and technical objectives.

If you are migrating to EpicCare, remember that it is patient and encounter-centric. Therefore, use patient, encounter and content-centric data design.

Interface design:
Hayes Management WHITE PAPER: Data Conversion Best Practices

- **Patient Demographics** – You’ll need to determine the trusted source of your demographic data. This may be your legacy EMR, or it may be your ambulatory registration system. You will want the most up-to-date source that can be matched to the clinical data converted from the legacy EMR. You’ll also want to consider how new and updated registration data will flow to the new EMR in real time once the initial registration conversion is complete.

- **Encounters** – EpicCare is encounter-centric. It is our experience that most legacy EMR systems are not. Ultimately, what that means is that all chart data that gets loaded into EpicCare is associated with an encounter or “visit.” Patient contact or visit entries from the legacy EMR will not necessarily be an exact match to those in EpicCare.

- **Appointments** – Again, for those converting to EpicCare, it is important to utilize the smallest number of encounter types during the conversion load, because the clinical data conversions that will follow the scheduling load will attempt to link the visit entries to the encounters loaded by the scheduling load. Also, during the clinical conversion, historical data from certain document types will be mapped to create specific encounters. For example, a phone encounter will not attempt to perform encounter matching and should always create a new encounter.

  Future appointments that are converted or that are sent to EpicCare via a real time interface will not create an encounter when they are loaded to EpicCare as an open appointment. The encounter is created when the patient checks in. Generally, an appointment conversion table that maps visit types to encounter types is set up to identify what encounter type will be created when the patient is checked in.

**Clinical content**: This is the real essence of the patient chart and these conversions cannot be considered final until you have made decisions about the chart presentation that your clinicians are expecting. What tabs will appear in your chart review section? Which documents file to which tabs? What is the design of the print groups that display the clinical documents? Your organization could decide that specific transactions will not convert effectively and choose to not convert selected records and use abstraction to complete that portion of the chart.

**Stakeholder sign-off**: It is important to get verification that technical design adheres to functional and clinical design needs. Get stakeholder sign-off on everything.

**Phase IV – Test**

As requirements are finalized and the execution of the build phase progresses, the testing phase begins. This includes unit, data communications, integration and user acceptance tests. Results are evaluated against the agreed-upon success metrics and validation criteria and should continue until a full volume conversion is tested. All defects discovered during the testing phase follow the appropriate remediation process.

During the testing phase, site staff should begin training on interface management and transaction remediation.
Best Practices: Test

Full volume testing: Work with technical support to plan disc space. Build your testing schedule using estimated process times. Perform incremental tests at increasing volumes up to full volume.

Test environments: You should have at least two: a primary test environment for incremental volume tests, and a secondary test environment for simulated production. All build updates must be promoted to these environments. Environment refreshes may be required.

Tracking: Good testing requires good tracking. Use tracking tools to monitor testing progress, defect countdown, issue resolution, etc.

Clinician sign-off: Testing is not complete until the clinicians sign off. Be unrelenting about quality in testing. Use an appropriate mix of electronic and human validation. (Electronic validation is optional. Human validation and signoff by clinical SMEs is essential.)

Phase V – Go-live

There are two go-live options: “rollout” and “big bang.” The go-live approach will have been determined and planned in the requirements phase. The big bang conversion can be more challenging than a rollout conversion in terms of training and support resources. However, a rollout approach is technically more complex.

Each go-live approach includes:

- Deployment
- Cutover support
- End user support
- Issue documentation and resolution
- Validation that the desired project outcomes were achieved

Best Practices: Go-live

The go-live is where all the work on build, conversion design, data transformation and interfaces come together – and in a short time frame. The legacy system is going down and the new system is coming online. Data must continue to flow correctly. Conversion activities are completed just moments before interfaces are started and the system go-live released to the initial users. It is an intense time for the project teams and tight coordination is essential.

Bulk and gap conversions: Bulk conversions sometimes take days to complete. As a result, a smaller bulk conversion is needed, often called a “gap conversion, because it picks up the data while the bulk conversion is running and helps sync up conversions to the moment of the go-live. Sometimes there are multiple gap conversions that cover incrementally shorter time periods (six weeks, two weeks, two days) to sync up conversions to the go-live moment.

As the final gap conversion ends, the interfaces must be started to assure that no transactions occur between the end of the conversions and the start up of the interfaces.
Big-bang vs. rollout: If the go-live approach is a big bang, the legacy system must be locked out to prevent any new transactions once the final gap conversion is complete. If the go-live approach is a rollout, there must be real-time conversion interfaces that transfer all new manual activity in the legacy system to the new EHR. These real-time conversion interfaces will be based on the bulk conversion designs, but must be designed specifically for this transfer of data during the rollout. Just as with all other interfaces, these must start as soon as the last gap conversion ends and then must continue to run until the rollout is complete and input to the legacy system is ceased.

Summary

A data conversion is a complex choreography of clinical, technical and strategic moving parts. It requires a methodology and the execution of best practices that ensure its cohesion and success. Data conversion is the key to a successful EMR migration, which can provide a healthcare organization with myriad opportunities for better outcomes – clinically, operationally, and financially.

About Hayes

Hayes works with healthcare organizations to increase net revenue and improve patient experience. Hayes is ranked Top Professional Services Firm by KLAS,* and has received multiple Best in KLAS awards since 2005. Hayes is also ranked in Healthcare Informatics’ Top 100 and Inc. 5000.

Hayes consultants are subject-matter experts in IT strategic planning, revenue cycle improvement, system implementation, interoperability and business and clinical operational efficiency. Hayes also offers software solutions to improve efficiency and productivity. To learn more about Hayes’ services, visit www.HayesManagement.com or call us at 617-559-0404.