Clinical Data Conversions Best Practices

Empowering Extraordinary Patient Care

Agenda

- **Our goals for today**

- **An introduction**

- **Scope**

- **A selection of best practices for your Healthcare IT system conversion projects**

- **An opportunity to ask your questions**

Goals

- **Focus on healthcare IT information system migrations**

- **Remain agnostic in regards to vendors and applications**

- **Share practices that are important to the clinic but are sometimes overlooked.**

- **Share not only *what*, but also *why***

Introduction

Michael Tamlyn, Integration Architect

- **Professional and enthusiast software developer**
- **6+ years of working in health care IT**
- **Integrations, conversions, and tool development**
- **If you are an Allscripts client, you may have used some of my software.**
  - Enterprise Interface Tools

Scope – What is a conversion project?

- **Extraction of clinical data from a data source**
- **Transformation of the clinical data**
    - Filtering
    - Translating
    - Scrubbing
- **Loading of the clinical data into a data store or application**
- **Not real-time or triggered**
- **Executed only once or a small number of times**

Scope – What is a conversion project?

- **Conversion projects are**

    – Individual

    – Complex

    – Logistical

    – Technical

    – (Sometimes) Emotional

Scope

- **A selection of practices out of many**

- **For each practice**
  - A description of the practice
  - Why the practice is important

- **Practices can be implemented in many ways and varying scales, so we will not go in depth in answering the question *how***

The Practices

1. **Consciously decide on scale and purpose**

2. **Quickly provide access to data sources and testing environments**

3. **Develop iteratively and stay flexible**

4. **Create an audit trail that can be queried**

5. **Own your transformations and translations**

Best Practice #1

Consciously decide on scale and purpose.

Consciously Decide on Scale and Purpose

- **Why?**

    Often projects are scoped without knowledge or consideration of all the available options

Consciously Decide on Scale and Purpose

- **Define the real goals of the project**
    - Transition to new application
    - Improve Workflow
    - Clean-up legacy data
    - Remove duplicate charts
    - Legal compliance

- **Spread understanding with your team throughout the project**

Consciously Decide on Scale and Purpose

- **Consider all of your options**
  - Discrete vs. Non-Discrete Data
  - Data Exports and Archives
  - Data Filters

- **Each data type may require a different choice of options**

Consciously Decide on Scale and Purpose

- **Consider the limitations of your data source and legacy applications**
    - Size
    - Location, accessibility
    - Quality
        - Original Method of input

- **Scale is limited by the quality and accessibility of the source data**

Consciously Decide on Scale and Purpose

- **Legal Compliance**

  - What rules are in scope?

  - Must version history of data be included?

  - Must ad-hoc notes and annotations be included?

  - How many years of history?

- **Some purposes are mutually exclusive**

  - Example: archiving for legal compliance and minimizing data scope to avoid information overload. Two separate conversions required.

Consciously Decide on Scale and Purpose

- **When scale and purpose are clearly defined, they inform every other decision the team makes.**

- **Provides a greater understanding of why choices were made, minimizing confusion and missed expectations.**

Best Practice #2

Quickly provide access to data sources and testing environments.

Quickly provide access

- **Why?**

  Often providing access to source data and test systems is handled only after a team member requires access to complete a task, delaying productivity

Quickly provide access

- **Access strategies and requirements should be identified during project inception**

- **Access affects scope, scale and timeline**
  - External vendors may need to be engaged
  - Lead-time may be required
  - Unexpected costs may be involved

- **Analyzing access early reduces risk**
  - Ensures you understand all available options.
  - Identifies potential bottlenecks
  - Helps define a realistic project timeline

Quickly provide access

- **Eliminates surprises early in the project**

- **Avoids unnecessary delays**

- **Ensures engagement of project team leads to immediate productivity**

Best Practice #3

Develop iteratively and stay flexible

Develop Iteratively and Stay flexible

- **Why?**

  Conversion projects and system migrations are inherently difficult to predict

Develop Iteratively and Stay flexible

- **There will be issues. Expect to make many small adjustments.**
    - Identify and address quickly
    - Not a sign of failure

- **Test individual data types when they are available, don't wait**
    - Work in parallel: test one, while developing another
    - Confirm conformance with specifications early
    - Provide feedback to analysts and developers on nuances and customizations

Develop Iteratively and Stay flexible

- **Trust your team**
  - They want the project to succeed
  - They will naturally push for the "right way"
  - Listen when your team feels something is out of scope or that a decision will increase project risk

- **Recognize the limitations of the source data and target systems**
  - Features sets
  - Models
  - Workflows
  - Free-text vs. enforced data entry

Develop Iteratively and Stay flexible

- **Make continuous progress**

- **Keep team working in parallel**

- **Handle change more gracefully**

- **Reduce risk to your go-live date**

Best Practice #4

Create an audit trail that can be queried

Create a Queryable Audit Trail

- **Why?**

  Without an audit trail, data will be loaded without a reliable method to track it back to the source

What is a Queryable Audit Trail?

- **Tracks what you did and when you did it**

- **Records what the results were**
  - Actions, identifiers, errors

- **Can be analyzed quickly and reliably**
  - SQL, XML

- **Not an application log**

Why Queryable?

- **Standardized reports can be defined and reused**
  - Patient matching
  - Dictionary mismatches

- **Effort to create ad-hoc reports is minimized**

- **Reports can be tweaked and modified with little effort**

- **No data is skipped**

- **With SQL, queries can join with data in other databases**

Create a Queryable Audit Trail

- **Helps answer common testing and post-live questions**
  - Why does this data appear this way?
  - Why was this data added to this patient?
  - Why doesn't this patient have the expected number of clinical items on his or her chart?
  - Why did this data get duplicated?

Create a Queryable Audit Trail

- **Investigating issues without an audit trail is difficult**
  - Time consuming
  - Unreliable
  - Involves more resources than necessary

- **Audits can help you perform gap conversions**
  - Know what data was successfully loaded
  - Allows overwrites/updates to be verified.

Create a Queryable Audit Trail

- **Quickly understand what actions were taken, what their result was, and why**

- **Generate reports to spread understanding within the team and to stakeholders**

- **Improves your confidence with the results of your project**

Best Practice #5

Own your transformations and translations.

Own your translations and transformations

- **Why?**

  Often translations as simply a task to be completed as quickly as possible, resulting in time consuming issues during testing and go-live.

What are transformations and translations?

- **Transformations are any time you are modify the source data before loading it.**

- **Transformation are done either ahead of time, or in-line during the load process**

- **Examples**

  - Translation tables (most common)

  - Clean-up

  - Duplicate removal

Own your translations and transformations

- **Accept the scale of the task of defining translation tables**
  - Often the most time consuming task
  - The tables represent your source data. Small workarounds and shortcuts add up over years, sometimes results in large translation efforts

- **Recognize the importance of your transformations**
  - Can affect workflow
  - Clinicians should approve every transformation and translation
  - Poor transformations and translations can lead to patient safety issues

Own your translations and transformations

- **Minimize the effort required to define translation tables**
  - Technical resources should attempt to clean up garbage from data, if applicable
  - Only transform the source data you will load

- **Assign the right resources**
  - Assign enough resources to define translations comfortably
  - Assign the right resources so that definitions are clinically safe
  - Resources should be *available*

Own your translations and transformations

- **Ownership leads to understanding, so you only commit to the amount of risk and effort your team is comfortable with**

- **Improved data quality**

- **Reduced go-live issues
  Eliminate surprises**

Conclusion

Conversions are complex and transformative projects. It is you, as the clinic, who has to live with the results. Set yourself up for long-term success by demanding that these and other best practices are leveraged by your team and your vendors.

Questions?