Using the [HL7 v2.5.1 Implementation Guide for Immunization Messaging, Release 1.5](https://www.cdc.gov/vaccines/programs/iis/technical-guidance/downloads/hl7guide-1-5-2014-11.pdf)  and the [CDC Endorsed Core Data Elements](https://www.cdc.gov/vaccines/programs/iis/core-data-elements/iis-func-stds.html) (and accompanying [IIS Functional Guide, Vol. 2: CDC Endorsed Data Elements](https://repository.immregistries.org/files/resources/5a83216a1d369/aira_functional_guide_vol2_final.pdf)) as references, create a list of each “R” and “RE” field for each HL7 segment.

For each field, list possible error conditions for which a specific error message is warranted. For example, a required data element is missing, or it populated with an invalid value. “Invalid” could mean a coded value that doesn’t exist in the expected value set, or an improperly-formatted e-mail address, etc.

Also consider business rules for data quality as well as data consistency implications between a field and a different one in the message (e.g., PID-7 date of birth post-dates the RXA-3 administration date), or even data already in the IIS (e.g., a “delete” message that doesn’t match to any existing record in the IIS).

Assign priorities to those error conditions – these priorities indicate to what degree we expect IIS to generate error messages with specific error codes. Nathan’s guidelines:

* HIGH: A concept that the IIS should be checking for and indicating as a problem in most (if not all circumstances) and in every jurisdiction. Items should only appear on the list that we are confident and IIS should say something about this. Good example: Sending "DOB MISSING" in PID-7. Every IIS in the US should be able to identify this problem and say something about it. But we are still steering clear of saying it must be an E Error. This list will often be what are usually errors but can include things that may be warnings as well.
* MEDIUM: A concept the IIS should probably be checking for and sending at least a warning back. Many of these will be warnings, or could be errors depending on IIS policy. But strictly speaking the IIS wouldn't be expected to necessarily be messaging back. Good example: Address is missing.
* LOW: Everything else. This is the "rummage bin" of the issues. We expect that most of the issues will fall in this category.

Review the existing [Compiled Error Codes](https://repository.immregistries.org/files/resources/59ee748913785/compiled_error_codes_20180320_update.xlsx) to determine if there are already defined error codes for each error condition. We suggest focusing on whether the existing codes capture the specific field / error condition in question. If the only applicable existing code is a generalized code not specific to that field (particularly codes 1 through 7), then consider creating a new code.

When creating new codes, first attempt to classify the error condition using the existing scheme in the existing [Compiled Error Codes](https://repository.immregistries.org/files/resources/59ee748913785/compiled_error_codes_20180320_update.xlsx) list – this will determine the first 2 digits of your new error code. For example, the 20XX series codes are for conflicting data error conditions, the 21XX series codes are for inappropriate dates, etc.

NOTE: We recommend further research and discussion about the classification system for error codes. For example, the classification system could potentially be used to automate the notification routing for errors to the appropriate actor who should be taking additional action. Along those lines, are the existing classifications too broad? Too specific? What is the adoption level of the existing [Compiled Error Codes](https://repository.immregistries.org/files/resources/59ee748913785/compiled_error_codes_20180320_update.xlsx), and what flexibility is there to re-classify existing codes without breaking existing implementations?