LOINC Mapping: Maintenance

Lessons Learned

The mapping of the Laboratory Information System (LIS) test catalogue to the pan-Canadian LOINC Observation Code Database (pCLOCD), once completed, requires ongoing maintenance to keep the maps current. The pCLOCD is published in Excel or Access format, it is expected that mapping projects have access to a mapping tool (LOINC Mapping: Planning). The pCLOCD and the parent standard, LOINC, provide updated releases twice a year.

Considerations:

- Is there an established maintenance cycle set up to support the mappings?
    - Is there a process in place to support out of cycle changes?
- Is there a change management process that requires submission of changes made in the LIS and any upstream systems such as EMR's?
- Have all the internal and external resources to support changes and new releases been identified and secured?
    - Are vendor resources required to apply any of the changes identified?
    - Are there charges associated with vendor support?
- Is maintenance being managed in a decentralized or centralized model?
    - If decentralized, is there a process in place to ensure changes were not overlooked between sites and were applied consistently between LIS maps?
- Are the published releases from the Standards Development Organization's (SDO's) understood or is additional training, resources required?
- Is each map supported with an audit log that keeps track of changes to the map?
    - Is documentation required to support changes, deprecations and code promotions?
        - If so, where is the documentation maintained and do the right people have access to it?
- Are the data elements utilized in the implementation provided only by pCLOCD?
    - Has additional metadata been added or pCLOCD data been modified (i.e. interface terms that are not part of the published artifact, overwriting existing viewer names, extension codes)?
- Can the codes that are in use by the implementation be identified easily?
- Has a maintenance checklist been created?

Is the architecture of the repository understood well enough to analyze changes that may affect historical information?

Recommended Actions:

- Maintenance cycles should be established early in the project life cycle.
    - Consider creating a road map that identifies new releases and associated timing as well as providing windows for other changes to occur.
    - Maintenance changes usually involve a number of resources (internal and external) and need to be planned for accordingly.
    - Most SDO's publish new releases of their terminology biannually. Although it is not imperative to update systems with each release, each one will need to be analyzed for impacts to current mappings. Out of cycle changes will also occur if local extension codes are being submitted for consideration to the parent SDO.
- LIS systems tend to change frequently. To maintain the integrity of the repository and other destination systems, changes must be considered as soon as they occur, and schedules put in place to implement those changes. Implementation project should ensure that a request for change process is in place early with all appropriate stakeholders. This process will be used to accept and distribute changes that are made to maps and associated codes and local tests.
- If repositories are sending lab reports to upstream systems such as EMR's, a change management process should also be in place to support receiving changes and submitting changes to those systems.
- Timing of the changes needs to be managed to make sure that as each system changes it does not cause a subsequent error in the upstream receiving system.
- Resources that are required to support changes and new releases may include system administrators, business analysts, Information Technology /Management (IT/IM) analysts and vendors or contractors. Ensure the costs for changes that involve vendors or consultants are accounted for, especially if it is anticipated that changes will be implemented and billed as they occur. If costs are charged by occurrence rather than lump sum, this could become quite expensive.
- Systems administrator or IT/IM resources will be required to:
    - Get access to the new release;
    - Upload the new release to the development environment of the tool (if applicable), ensuring adequate backups etc. are in place prior; and
    - Create and run queries and provide reports.
- Business analysts/system administrators will be required to:
    - Get a list of codes in use in each map;
    - Identify queries to look for changes that impact maps;
    - Analyze reports for impacts to maps;
    - Review new codes to see if they address any mapping problems or are similar to existing extension codes;
    - Do any remapping required;
    - Implement approved changes into the production environment; and
    - Implement changes in the LIS.
- Vendors/Contractors/IM/IT resources will be required to:
    - Implement changes in the repository and any upstream systems (EMR's etc); and
    - Run associated Quality Assurance processes and Conformance testing as required.
- If possible, look for ways to consolidate maintenance changes across different LIS' if maintenance is being managed in a decentralized model. This will help ensure that changes were not overlooked and are applied consistently between LIS maps.
    - If maintenance is managed centrally then all of the changes can be analyzed at once and a list of changes created and communicated to the resources from each LIS or EMR etc. along with the implementation schedule.
- Not all release artifacts are supported by extensive release notes. If release notes are published, ensure they are available and provided to all the resources that may need them.
    - Allow sufficient time after receiving the release notes to review and ask any questions prior to the start of the update.

- Each change that is made to a map, especially a map that is used in a production environment, should be documented along with supporting information that explains the change. This information should be kept in close proximity to the actual map if possible.
    - The audit log should keep track of the date the map was effective and the date is was inactive as well as any changes to the code that is associated with a map.
    - If a code is deprecated and replaced with a new code, it is beneficial to include a note along with the map that indicates why the map was changed.
- If the implementation has added metadata to the map ensure the update process does not overwrite the data when the new release is loaded into the system.
- Biannual releases of pCLOCD/LOINC need to be compared to the codes in use.
    - Changes to codes not in use will have no impact on maps and may only need to be analyzed for other purposes.
    - It is important to generate all queries based on codes in use and those codes need to be easily identified for this purpose.
- A checklist is required to ensure all changes have been analyzed for impacts to existing maps. Ensure the checklist contains procedures to support the following:
    - Periodically LOINC codes change the component name, typically to add clarity or to correct a component that was misrepresented. This change is likely to impact the map. (i.e. A LOINC code changed from 'methodless differential' to 'manual differential' which significantly changes the use of the code. The historical records needed to maintain the original meaning of 'methodless differential' but any new records needed to be remapped. In this scenario it may be impossible to maintain two meanings for one code and therefore future use of the changed code is prohibited. An extension code would need to be created for future mappings of both 'methodless differential' and 'manual differential').
        - Anticipating this type of change early in project planning allows sufficient time to create processes to support a change of this magnitude.
    - Each release contains codes that are deprecated. Implementations are encouraged to deprecate the use of the code and remap to the replacement or an alternative code.
        - This change may have impacts on trending, graphs and any other statistical analysis that is relying on the unique identifier of the code.
        - Consider looking for an opportunity to build in relationships between codes that would support this process.
    - New codes are added with each release.
    - New codes can have two flavors; the first is the new code that originated from the SDO release and the second is the code that was created as a result of the submission of an extension code by the implementation. New codes that are created as a result of the submission of an extension code by the implementation require additional considerations:
        - Making a change to the unique identifier is a significant change to a code and to the mapping relationship and should be planned out carefully to make sure all possible implications have been considered both upstream and downstream from the repository.
        - There are at least two models to consider to support the conversion of an extension code to a new code;
            - The first is to add the new code, deprecate the local extension and remap.
            - The second is to update the local extension with the new code identifier, which by default creates a new mapping relationship between the identifier and the local test.
            - In each model ensure that there is a way to associate the local extension to the new code similar to the association created for a deprecated code.
- All changes that are done to existing maps need to be considered in light of what exists already in the repository and in any upstream systems.
    - There are many ways to query information from a database, and if a key requirement is based on the LOINC code for a specific purpose, the work will be impacted each time there is a change to a map that is not supported by cross relationships or supporting documentation.

Get Involved:

Join a collaboration group to solve interoperability challenges.

The Health Terminologies Community is a open forum for sharing and communicating on topics of terminologies and classifications and their use in Canada. We welcome all participants.