Technical Roadmap

What is the OpenMRS Technical Roadmap?
The Technical OpenMRS Roadmap is a set of milestones for our Platform, Reference Application, community-sponsored modules, and related tasks that help us meet the needs of our implementations.

For information about how the roadmap milestones are chosen and prioritized, see the Technical Roadmap Planning page.

For details of recent releases and release notes, see Releases.

Milestones

2021 Goal Brainstorming

Process Ideas:

Structure based on resources: What Implementers are doing and have near-ready, what we can reasonably expect from volunteers

Talk: Can we create a Label in Talk that anyone can apply to conversations that should inform our technical roadmap?

Actual Tech Priorities (Ideas):

- **Platform QA**: Some regressions - need more automation to catch in advance, esp. build into PR pipeline
  - Automation +++: Get platform to point where enough is automated so have confidence in quality and release more frequently
- **Simplifying Deployment**
- **Support for timezones** across the Platform (and beyond, see also about HFE here.)
- **FHIR**
  - Support for FHIR Group through maybe a new Core Group entity replacing Cohort (see Slack discussion here.)
- **Metadata Sharing**: a solution that meets majority of needs (if Iniz, maybe expand and handle a few additional use cases; come up with something that's better than our old tech for metadata sharing)
- **Integration**: Clarify actionable things we can do to better integrate into other Digital Health tools that are typically found around an EMR like OpenMRS.
- **i18n**: Fully internationalizing metadata concepts (but is this really a priority, pain point?)
  - Support for i18n of metadata in Core (see for example this thread.)
- **Download & Run**: Easier to use without getting bogged down in details - something non-tech end user can work with. Standalone hasn't advanced in a while - needs love for feature updates or different approach or make useable again (e.g. some Mac versions no longer running). People using Standalone in production b/c simple to manage, came up with backups automation approach. So easy to download, click and it runs. Using tech that's already available on users' computers.
- **Sync**: Most implementations not connected to single server; most combine, do analysis later. Is this a priority for implementers now?
- **Patient Lists**: FHIR-based patient lists proposal from Ampath
- **Library dependencies between modules** (not high priority because so time consuming, and there are (ugly) workarounds)
- **Decision Support**: Some meaningful step (e.g. with simpler ETL solution: flattened data, generates table, SQL, extrapolate from that). Feels we are waiting for AES to eventually address.

Platform 2.5 (Q4 2021)

- **Support for Tomcat 8.5+**
- **Support for FormRecordable for more than just Obs and Condition**
- **Administration via REST**

Tickets for Platform 2.5
Additional candidates for 2.5:

- Patient Lists (Cohort Module changes)
- User settings
- Moving OrderTest toward ServiceOrder to support Referrals

Reference Application 2.12 (Date TBD) - Theme: TBD

TBD

Release Calendar

How to update this calendar

List View

Team Calendars

Calendar View

Team Calendars

Someday

Reference Application Someday

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Status</th>
<th>Point of Contact</th>
<th>Comments</th>
</tr>
</thead>
</table>

9 issues
<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Status</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add Order Entry UI</td>
<td>Add some sort of user interface for doing order entry.</td>
<td>Daniel Kayiwa</td>
<td>We could polish up the order entry module, or do an OWA from scratch. Whichever we find easier.</td>
</tr>
<tr>
<td>Add Patient Flags Module</td>
<td></td>
<td></td>
<td>Still under development.</td>
</tr>
<tr>
<td>Responsive ability to the clinician facing dashboard</td>
<td>RA-1285 - Getting issue details...</td>
<td>Accepted</td>
<td></td>
</tr>
<tr>
<td>Core and modules to advertise capabilities that can be configured and manipulated</td>
<td>RA-1292 - Getting issue details...</td>
<td>NOT DONE</td>
<td>Not started</td>
</tr>
<tr>
<td>Pre-built Reporting Tools</td>
<td>This includes the ability to do several things. 1. Run a report for a patient directly form the patient dashboard.</td>
<td>IN PROGRESS</td>
<td>Patient summary module does not provide adequate configuration and doesn't work in Ref App. 27-March: We could include Reporting REST documentation that Darius Jazayeri and the Andela team have done.</td>
</tr>
<tr>
<td>OCL subscription module</td>
<td>Using KenyaEMR as a use case, create a tool for subscribing an OpenMRS instance to a dictionary (e.g. the CIEL dictionary)</td>
<td>BETA</td>
<td>Nicholas Ingosi and Rafał Korytkowski - OCLM-24 - Getting issue details...</td>
</tr>
<tr>
<td>Retrospective data entry</td>
<td>Basic support for retrospective data entry within the Reference Application</td>
<td>NOT STARTED</td>
<td>Needs a user centric story - add to existing ticket.</td>
</tr>
<tr>
<td>OpenMRS Web Framework</td>
<td>From discussions in #MOZ15, we would like for the OpenMRS Platform to evolve toward providing a web framework that allows developers to add functionality using standard development tools (e.g., HTML5 + JavaScript against REST services, AngularJS apps, OWA, ...).</td>
<td>IN DESIGN</td>
<td>Burke Mamlin - Ranking REST tickets on 5 Aug design call. Burke Mamlin and Darius Jazayeri discussed 24-July-2015 and Darius will make a Talk topic to move this forward. Bahmni technical deep dive scheduled for 4 June Developers Forum. 17 June design forum will discuss progress (coordinating various efforts). As well as look at how to make REST services more robust. Darius still working w/ Bahmni on fundamental pieces of their web framework to pull into OpenMRS (will talk about this on future call). 25 June dev forum on REST web services: how to substantially improve from W. Burke Mamlin &amp; Darius Jazayeri.</td>
</tr>
</tbody>
</table>
OpenMRS has a lot of flexibility and extensibility with a central concept dictionary, RBAC, forms, reports, modules, and apps; however, it’s not always easy to know which metadata goes with which functionality. The goal of vertical packaging is to define best practices for managing and relating all of the components (metadata & behavior) that work together to solve a particular problem within OpenMRS. Eventually, we envision a way that someone could easily add the MDRTB package to their OpenMRS implementation to begin treating MDRTB patients... or upgrade their Oncology package, etc.

**Condition List**
Manage & view patient problems (e.g., on the patient dashboard and integrated with diagnosis capture)

- **EA-40** - Getting issue details...

See also:
- **RA-209** - Getting issue details...
- **RA-580** - Getting issue details...

See Condition List board

**Basic Order Entry for meds and tests**
Basic ordering of meds and tests "out of the box" in Reference Application.

**Ad Hoc Analysis tool (v1)**
Incorporate new cohort definition tool.

- **RA-261** - Getting issue details...

**Concept Management Improvements**
Allow for concept merging and easier browsing through concepts and references terms without losing frame of reference.

- **RA-341** - Getting issue details...

**Provider Management**
Would include provider types and ability to retire the old provider management module. Will remove UI library module once provider management is in the core.

<table>
<thead>
<tr>
<th>Platform Someday</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Feature</strong></td>
</tr>
<tr>
<td>Anonymous Patients</td>
</tr>
<tr>
<td>RA-62 - Getting issue details...</td>
</tr>
<tr>
<td>Test Patients</td>
</tr>
<tr>
<td>RA-65 - Getting issue details...</td>
</tr>
<tr>
<td>Clinical Encounter</td>
</tr>
<tr>
<td>Patient Lists</td>
</tr>
<tr>
<td>Program Enrollments</td>
</tr>
</tbody>
</table>