[Burke Mamlin](https://talk.openmrs.org/users/burke) @ OpenMRS on Prioritizing the Roadmap

<https://talk.openmrs.org/t/how-implementation-needs-get-prioritized-on-the-roadmap/6082/14>

It might also help to know some of the history that has been touched on in above comments:

* **Survey-based Road Map updated 1-2x/year.** Our early attempts included surveys and at least an annual (sometimes twice a year: once at our implementers meeting and once at a leadership meeting), we would survey as many implementers as we could to get their top priorities ahead of time and at the meeting we'd turn them into a road map.
  + Pros
    - Everyone had a chance to contribute
    - It got semi-regular attention and we had an informed road map
  + Cons
    - Implementation needs & priorities changed far faster than a 6-month period (sometimes weekly or monthly), so by the time we were a few months into our "road map," we could be working on things that were no longer a priority.
* **We turned to voting.** In an attempt to address the moving target of priorities, we turned to vote-based prioritization a few years ago and created [this wiki page](https://wiki.openmrs.org/x/3AAdAg) to summarize new & most-voted issues (now broken since it uses outdated macros).
  + Pros
    - Anyone could vote
    - Priorities could be based on true need
  + Cons
    - People complained that voting unfairly favored larger organizations (more people to vote on issues)
    - It was harder to organize meaningful/themed sprints and development felt "rudderless"
* **Road Map Committee.** To try to compromise between a 1-2x/year road map and a purely vote-based road map, I tried to get people to participate in a "Road Map Committee" with the idea that we'd meet more regularly and create a process that was informed by surveys, votes, and whatever formula/process we could invent to drive a responsive road map in a fair & agile manner.
  + Pros
    - Tried to combine the best parts of our prior attempts
  + Cons
    - Nobody participated.
* **Weekly Project Management.** After the road map committee approach flopped, we landed in a place where leadership wasn't regularly polling implementations (we still tried to do this in occasional Dev Forums), we weren't satisfied with purely vote-driven prioritization, and, for a number of reasons, the number of people invested in driving a road map for the reference application was declining (e.g., PIH, AMPATH, Kenya-EMR, Bahmni, and others using various flavors of front-ends). So, development was, once again, rudderless. In response, I set up our weekly Monday Project Management call, which has a handful of regular attendees, occasional attendees representing release management or specific implementation needs. We've used this to manage our current [technical road map](http://om.rs/roadmap).
  + Pros
    - Provides at least *some* oversight and direction for development
    - Uses the wiki to communicate a [technical road map](http://om.rs/roadmap)
  + Cons
    - Limited participation, so not fully representative
    - Focused more on technical project management, so doesn't meet all implementation needs nor the "solution" for community prioritization.

Given the above, a twice-per-year implementation-driven prioritization would give us back what we were getting when developers & leadership surveyed implementations and would be extremely helpful in informing a technical road map, but (for the reasons & history above), I'm not convinced that a twice-a-year process can fully suffice.

Ideally, we'd have a well-communicated process informed by multiple factors and updated frequently (at least monthly):

* **Implementation needs** (e.g., regular prioritization meetings & surveys, votes on tickets, etc.)
* **Strategic needs** (e.g., considering the long-term vision, partnerships, etc.)
* **Available resources** (e.g., high priorities that lack resourcing may move slower than lower priorities that are fully resourced or a lower priority need that fits neatly within the timing & scope of a GSoC project)
* **Development needs** (e.g., a high priority need that is poorly defined may take a back seat to sprinting on a lower priority need that has a theme + tickets ready for work + BA support)

As I mentioned above, coming up with a process like this for the Platform is part of our objectives in our operational plan. How well we can use implementation priorities to drive a web application (reference application or community distribution) will depend on the extent that we can align the community on web-based conventions (a separate conversation I will be pushing for imminently).