

Tri-council recommendations for:

- Initial introduction of applications into FOLIO
- Maintenance of dual Eureka and Okapi community environments
- Defining applications and a proposed ideal organization of current modules into applications
- Updates to technical council and product council module review processes
- Collaborative practices in the maintenance of applications
- Changes to release management

NB: The LC/ 'Main line FOLIO' context

EBSCO will introduce applications as part of their work for the Library of Congress prior to the introduction of applications into the larger FOLIO community. The recommendations here apply to the use of applications in the wider FOLIO community and not the initial work for LC. There is currently uncertainty about what the exact state of application organization will be at the time of introduction to main line FOLIO.

Initial introduction of applications into FOLIO

- “Applications” refers to the bundles of modules described by application descriptors. “Apps” refers to functionality in FOLIO that is accessed by clicking on the icon for a given app. These terms are closely related and may cause confusion but reflect current community usage.
- One release prior to the introduction of applications into FOLIO, the shift to applications should be publicized in the community, particularly with dev teams and product owners. For example, if we wish to introduce applications in Trillium, the announcement should be made prior to the Sunflower release scope deadline in August 2024.
- After applications are introduced into FOLIO, all modules, including new modules, must be part of an application. In the example above, all modules that are part of Trillium would be part of an application, including new modules.
- It is understood that at the time of the initial introduction of applications into FOLIO, existing modules will not be well-organized into applications. Further organization of applications will be iterative and require work by development teams. As this work proceeds, proposed organization of modules into applications should be presented to the product council on a per application basis for feedback from the community
- The proposed initial organization of applications should be presented to the Product Council as soon as possible

Maintenance of dual Okapi and Eureka community environments

Both Okapi and Eureka community environments will need to be maintained for a period of time. The number of environments of each type need not necessarily be equal and the number of Okapi environments will likely decrease with time. However, Okapi should not completely disappear from the community environments until:

- The Technical Council has approved the overarching Eureka RFC
- Documentation has been produced that is sufficient for system operators beyond the Eureka and Kitfox teams to set up a Eureka environment
- Two flower releases with Eureka as the primary platform for FOLIO are in General Availability (GA)

Defining applications

The right-sizing of applications is a key part of making the introduction of applications successful. Applications should group together enough modules that deploying FOLIO with applications is simpler than deploying FOLIO without applications, but applications should not group together so many modules that implementation decisions by libraries to adopt specific functionality are overly constrained.

Recommended principles for organizing applications:

- An application should be a set of modules that delivers a complete set of well-defined functionality in FOLIO
- In general, there should be a bias toward adding new modules to new applications rather than to existing ones
- Every effort should be made to define applications along functional boundaries, and not as reflections of the boundaries of development organizations
- Modules that deliver functionality that is part of multiple functional areas should be in their own application, rather than choosing between two functional applications. For example, organizations functionality is part of both ERM and finance functionality, therefore organizations should be its own application.
- Modules that deliver functionality that could reasonably be deemed unnecessary to an application should not be part of that application. For example, the storage of MARC data is unnecessary for the functional use of inventory and therefore the SRS MARC modules should not be part of inventory.
- An appendix with a [proposed organization of existing modules](#) is attached

Updates to product council and technical council review processes

- Technical Council should continue the review of individual modules and expand its processes to include review of existing modules as they are organized into applications. Modules moving from one application to another should be re-reviewed.
- Product Council should continue to review new functionality and should pay particular attention to the principles for organizing applications and be prepared to gather and give feedback from SIGs and other interested parties on the organization of applications, particularly during the iterative work to organize existing modules into applications. Functionality being moved from one application to another should be presented to the Product Council.

Collaborative practices in the maintenance of applications

- Application descriptors will be maintained by the team that releases the application
- Development teams who do not own applications should still be able to contribute to their development. Owning teams will need to take care to not release unfinished work as part of application releases.
- As modules move between applications, particularly in the iterative process of moving out of the initial application organization, teams will need to communicate if the creation of an application will result in leaving behind modules. For instance if four modules currently form an application and a team plans to create a new application from three of them, the team owning the fourth module should be notified as soon as the decision is made to do so. In this case, both teams will need to present application changes to the product council.
- There should be a process for the deprecation and sunseting of applications

Changes to release management

- Applications being released on independent schedules may require a new approach to testing, including a greater reliance on UAT of application releases rather than mass testing during bugfest. Until the initial large set of modules is broken up into applications, neither testing nor releases may change very much
- The community should monitor the organization of applications for a tipping point at which real change to release management can happen and movement away from monolithic releases might begin.

Questions still to be answered: