Development dependencies - Tips

Overview

A development dependency (for purposes of this page) is when a development effort requires requirements/work/support from more than one team. As FOLIO continues to grow, it is important that we have a common understanding for how to handle such dependencies. The key to successful handling of dependencies - Communicate as early and often as possible.

Ideally when to engage with POs and Dev leads on dependency: ~ By release scope definition deadline

Dependency scenarios

- Changing workflow (Example: )
- Shared apps/workflow (Example: )
- Shared backlog (Example: )
- Require development from another team (Example: )
- Developing in a module that is owned by another team (Example: )
- Developing in a module where multiple teams contribute code (Example: )
- New development team (Example: )
- Integration (Example: )

Best Practices for handling dependencies successfully

Still vetting/refining requirement.

- Requirements reviewed and refined with respective SIG(s)
- PO (with a dependency) > Please talk with the PO(s) that own the module/app/workflow. Provide the following details
  - High level overview of the change
  - Feature/User Story/Bug JIRA link/Flower release dashboard
- Outcome
  - Agreed upon next steps including who owns writing or implementing requirement and timing
  - Communicate <-> Collaborate.... Communicate <-> Collaborate

Team with dependency is planning/preparing for release to implement

- PO and Dev lead (with a dependency) > Please talk with the PO(s)/Dev lead(s) that own the module/app/workflow. Provide the following details.
  - High level overview of the change
  - High level introduction of the module/suite of modules for the developers who is to start working in these existing modules
  - Feature/User Story/Bug JIRA link/Flower release dashboard
  - Technical design
- Outcome
  - Technical design and approach alignment
  - Development efforts per team including timing for when dependencies will be ready
  - Testing approach (including test coverage)
  - PR reviews alignment
  - Definition of done alignment
  - Document agreements and implementation/testing plan
  - Communicate <-> Collaborate.... Communicate <-> Collaborate

During development and testing

- Jira epic/feature should link to all dependencies
- Schedule periodic check in meetings, e.g. respective scrum meetings (for efficiency)

Questions - please post your questions in this column

1. Is there a dependency? What if I’m not sure?
2. When multiple teams/POs work within a broad area, how do I know which team/PO to approach? (E.g. I need to do some work in Holdings related to Receiving in Acquisitions. I know who the MM POs are, but which PO and which team specifically?)
3. What if it’s a general area where no POs or teams exist, but which impacts multiple teams? (E.g. security)
4. What if my team is developing something that may be of use or interest to other teams and want to make those teams aware of this new functionality?
5. How do I prioritize the work with another team against the work set for my team?
6. What is the SIG’s role with dependencies?
7. What are developer’s responsibilities?
Provide an update on current state of development and testing
Participate in sprint testing / PO verification
Communicate <-> Collaborate.... Communicate <-> Collaborate

Before Bugfest
- e2e testing has been done before Bugfest (at least 2 weeks)
- Demo has been done that includes applicable SMEs/POs/QAs/Devs to ensure alignment (at least 3-4 weeks before Bugfest)
- Communicate <-> Collaborate.... Communicate <-> Collaborate

Ideas for communicating across multiple teams (please add your thoughts)
- The responsibility for communication and collaboration goes both ways <->
- Dedicated slack channel
- Recurring meeting
- Joint scrum meetings (once a month)
- Wiki page that is the source of truth and includes a decision log (when applicable)