

ERM Sprint 63 Retrospective

Date	22 May 2019
Sprint Report	https://issues.folio.org/secure/RapidBoard.jspa?rapidView=89&view=reporting&chart=sprintRetrospective&sprint=214
Sprint Log	ERM Sprint 63
Sprint Focus	Integration with Organizations

Participants

- [Claudia Malzer](#)
- [Gill Osguthorpe](#)
- [Jag Goraya](#)
- [John Allen Ferreira](#)
- [Kurt Nordstrom](#)
- [Mark Deutsch](#)
- [Steve Osguthorpe](#)

Apologies

- [Benjamin Ahlborn](#)

Agenda

1. Review Retro Board (10 mins)
2. Voting (2 mins)
3. Discussion (15 mins)
4. Summary Actions (3 mins)

Retrospective

Personal highlights

- More comfortable with project structure and processes
- Happy with the rhythm currently present in the project sprints
 - Nice to see this is being felt outside of development, in the feature / story preparation stages
- Better understanding of Grails and the project technologies, although at the expense of the first iteration taking longer.
- Code reuse between the 2 UI modules.
- Had time to pick up smaller tasks

What went well

- Feedback loop from testing to bugs being raised
- Transition from internal orgs to mod-organization-storage went fairly smoothly
- The management of the dev process seemed to be smoother.

What caused problems

- This project is often at bleeding-edge, and that can cause confusion as to where to find code or process that might be similar to what we want to achieve, to allow for reuse
 - We're doing Phase 3 planning at the moment, including looking at potential (future) module dependencies
 - We intend to flag these dependencies to other teams as soon as we have completed assessing the technical approach we intend to take
 - The Washington meeting (June 2019) is an opportunity to negotiate scheduling early development of potential dependencies
 - Improvements to release management for Stripes/UI modules may help mitigate a bit of the bleeding edge pain during delivery cycles
 - We're also looking at how to accommodate system analysis as part of the onboarding / knowledge share process
- Testing can be problematic when we rely on 3rd party plugins, eg, Find Org
- Onboarding time and learning curve for Grails and related techs we use are often high at the beginning of a developers involvement.
- JIRA is often not updated inline with the projects processes.
 - See discussion notes below

Discussion Notes

JIRA is often not updated inline with the projects processes

- The goal here is to
 - have accurate current status of progress and obstacles
 - reduce friction of Jira admin for all team members (including developers, testers, PO and scrum master)
- Leaving assignee unchanged helps make it easier to track work that has been 'owned' by a particular person
- This doesn't help identify who has responsibility for an item *right now*, although a combination of other fields may help
 - Tester Assignee field could be used to identify who is responsible for QA
 - Status of In Code Review should mean it's FAO Component Lead
 - Status of In QA should mean it's FAO Tester Assignee
 - Status of For Review should mean it's FAO Product Owner (the Jira Project Lead)
 - Status of For Release should mean it's FAO Lead Dev
- If a PR hasn't passed code review, the Jira issue status is not changed back to Open or In Progress - the re-work is managed through Github to-and-fro
- Automating Jira transitions from Github PR status could be done, if the integration has been set up.
 - Needs some verification
 - Automation will only be effective if only one PR is raised against each issue: exceptions would have to be manually handled
- Backlog / Kanban board filters in Jira have been set up to make it easier for leads and testers to identify items that need action based on status, rather than Assignee
 - Checking these should be part of daily routine

Still to consider:

- Inferring current state of progress based on Jira status still relies on developers picking up and (self-)assigning issues
- Initial issue assignment *could be* handled in Sprint Planning.
 - We do currently indicate who might pick up a piece of work in Sprint Planning log, but this isn't always reflected in Jira
 - Some items will inevitably be left in the backlog or arise during the sprint
- A story will often have front-end and back-end work.
 - These can (and should) be distinguished by separate subtasks
 - Subtasks can be separately assigned
 - Separate PRs can be raised against each subtask to support automated Jira transition
- But then, who should the story be assigned to? The PO?

Actions

- [Jag Goraya](#) to look into automated Jira transitions based on Github PR actions
- [Jag Goraya](#) to look at Jira filters for better visibility and tracking of current and previous work
- [Jag Goraya](#) to update [ERM Definition of Ready](#) to reflect use of Tester Assignee