Developer Communications Subteam
Geneva

November 4, 2019
Topics

• Pull Request Review Process
• Developer Communications
• Holding Repo Review / Promotion Process
• Branch Protections
• Missing Unit Tests
• Commit Messages
• Swagger Generation from Code
Pull Request Review Process

Problem - Pull Request Size
• PRs are submitted with too many lines of code changed making it difficult to review in a timely manner

Proposed Solution
• Break up the story to submit PRs with smaller chunk of code changes

Not Approved
Pull Request Review Process

Problem - Pull Request Detail

• Pull Requests do not contain the right level of information and should be reviewed by more than one reviewer with approvals by a minimum of two (2) people

• Pull requests should include a reference to the Issue #

Proposed Solution

• Use GitHub Pull Request templates
  • Template would include the basics (tests, multiple reviewers, comments, documentation)
  • Reference: Creating a pull request template for your repository

• Set branch protections on the repo to include >1 approvers
  • Holding repos should have branch protections set up so when / if the repo moves out of holding, the branch protections are in place once moved to main org
  • Written procedure followed by LF Release Engineers

Approved to use PR Templates
Not Approved to increase # of required reviewers / approvers
- reviewers need to be coders
Holding PR Review / Promotion Process

Problem – Holding Repo review process seeks to obtain TSC approval without proper consideration of what’s needed to actually move source code from holding to main org

Proposed Solution – Similar solution using Pull Request templates just for holding repos

"Brainstorming: can we have a small cut and paste template that goes into email with some of the required fields to be populated? I often assume that if a request is being made up to the TSC it has been vetted, but with Tony’s questions it is apparent that either sometimes the requests aren’t ready, or that it is not immediately obvious they are not ready. Not complete, but something like:

1. Requester: Working Group chair
2. Reviewers (min 2):
3. Black box testing passes: Y/N
4. Wiki documentation posted/updated: Y/N
5. Link to the relevant review criteria
   Example: https://wiki.edgexfoundry.org/x/_QHiAQ
6. Confirmation messages from the reviewers (conclusion of work)
7. Any concessions that have been made [the voting members should consider whether these should in fact block the move from holding to main org]
8. Whatever the definitions of done and ready are should be minimally pasted into the email request. Enabling someone like me, somewhat removed from daily engineering, to understand if a request is ready or not.

Note: DoD could include security validation, open source dependencies, licence and inclusion of attribution

Vetting within the WG meeting with formal nomination to move out of holding in TSC meeting
Branch Protections on all Repos

Problem – Missing Branch Protections

- Proposal to change >1 reviewer
- Approvers from different companies if possible
  - Addresses perception of a single company pushing through changes without any other reviewers
  - Sometimes there may not be reviewers / contributors from other companies so there’s no way to get around

Current Settings on edgex-go

Approved to set branch protections
Not approved to increase # reviewers
General

Problem - Missing Unit Tests

Proposed Solution

- Unit tests should be included in the code
  - We will use codecov.io to measure code coverage
    - Example: edgex-go
  - Pull Request reviewers should consider unit testing when completing a review before approving the PR
Commit Messages

Problem – Inconsistent Commit formats
• We should establish a more formalized and consistent format for all code commits

Proposed Solution
• Use conventional commits specification as outlined at
  - https://www.conventionalcommits.org/en/v1.0.0/
• Use tool named `git-chglog` to generate change log from commit messages based on type of change
  - Reference Example: https://github.com/edgexfoundry/app-functions-sdk-go/blob/master/CHANGELOG.md
• This will give us a human and machine readable commit message which will enable us to produce improved release notes

Bring Back to Architect’s Day for further discussion
Swagger Generation from Code

Problem – Manual way of producing Swagger documentation is disjointed from the code

Proposed Solution
• Annotate the code for automated swagger generation
• Use go-swagger

![GitHub Link to go-swagger]

[Code Snippet showing annotations for swagger generation]
Meeting Minutes

Problem - Working Group Meeting Minutes are not posted or are lacking detail of anything discussed within the meeting or action items that need tracked and worked

Proposed Solution
Ask for a volunteer during the meeting to take meeting minutes
- Post the meeting minutes within the next day following the meeting
- Take meeting minutes only to capture the key decisions and action items
Lack of Project Management Tracker

• Problem – Lack of a Project Management tool where working groups can see dependencies between working groups.

Proposed Solution
Use the GitHub Project Tracker and map Issues to cards that are processed automatically via the Project workflow automation.

Example: DevOps WG Tracker
Tracker affords Kanban style view where issues from multiple repos can be tracked in more than a single tracker.

Use tags to organize.