EDGE X FOUNDRY™

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

Not Approved

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

EDGE X FOUNDRY™

Pull Request Review Process

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

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 S

- 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

Holding PR Review / Promotion Process

Vetting within the WG meeting with formal nomination to move out of holding in TSC meeting

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)*

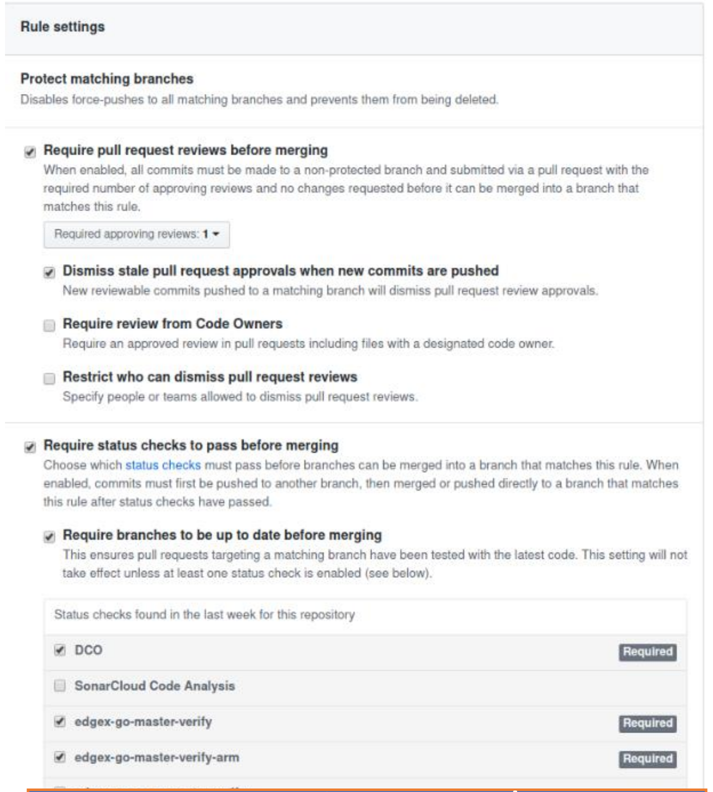
7. Any concessions that have been made [the voting members should consider whether these should in fact block the move from holding to main org]

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, license and inclusion of attribution

Branch Protections on all Repos

Approved to set branch protections
Not approved to increase # reviewers

Problem – Missing Branch Protections



Rule settings

Protect matching branches
Disables force-pushes to all matching branches and prevents them from being deleted.

- [✓] **Require pull request reviews before merging**
  When enabled, all commits must be made to a non-protected branch and submitted via a pull request with the required number of approving reviews and no changes requested before it can be merged into a branch that matches this rule.
  Required approving reviews: 1
  - [✓] **Dismiss stale pull request approvals when new commits are pushed**
    New reviewable commits pushed to a matching branch will dismiss pull request review approvals.
  - [ ] **Require review from Code Owners**
    Require an approved review in pull requests including files with a designated code owner.
  - [ ] **Restrict who can dismiss pull request reviews**
    Specify people or teams allowed to dismiss pull request reviews.
- [✓] **Require status checks to pass before merging**
  Choose which status checks must pass before branches can be merged into a branch that matches this rule. When enabled, commits must first be pushed to another branch, then merged or pushed directly to a branch that matches this rule after status checks have passed.
  - [✓] **Require branches to be up to date before merging**
    This ensures pull requests targeting a matching branch have been tested with the latest code. This setting will not take effect unless at least one status check is enabled (see below).

Status checks found in the last week for this repository	
✓ DCO	Required
☐ SonarCloud Code Analysis	
✓ edgex-go-master-verify	Required
✓ edgex-go-master-verify-arm	Required

Current Settings on edgex-go

- 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

Approved

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

Bring Back to Architect's Day for further discussion

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

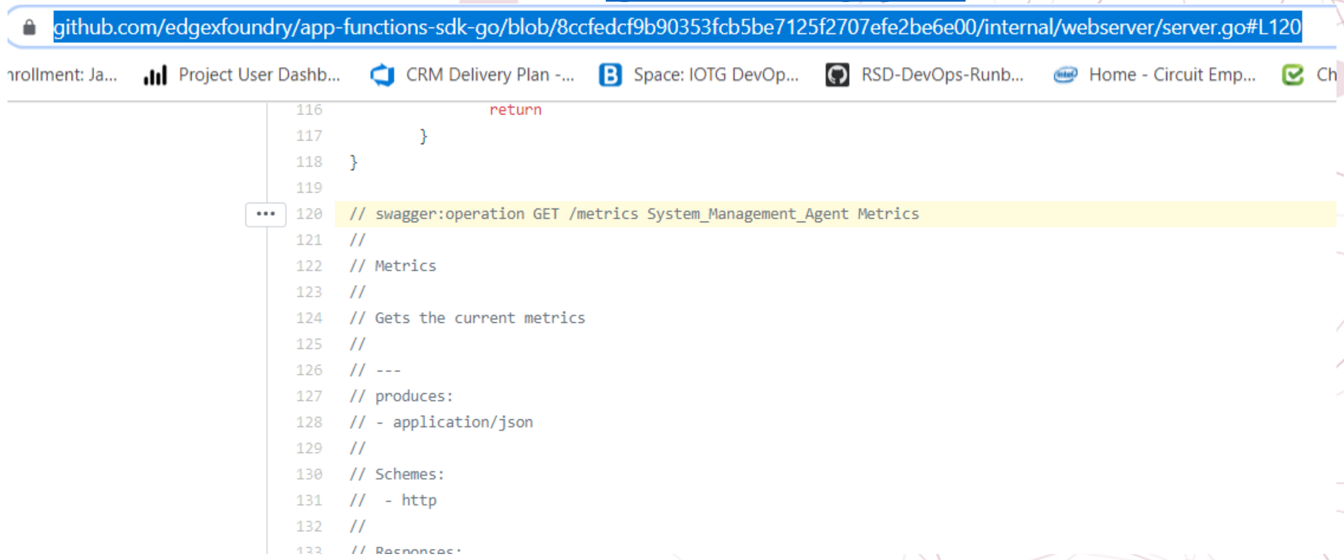
Swagger Generation from Code

Not Approved

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.com/edgexfoundry/app-functions-sdk-go/blob/8ccfedcf9b90353fcb5be7125f2707efe2be6e00/internal/webserver/server.go#L120
nrollment: Ja...
Project User Dashb...
CRM Delivery Plan -...
Space: IOTG DevOp...
RSD-DevOps-Runb...
Home - Circuit Emp...
Ch

```
                return
        }
}

// swagger:operation GET /metrics System_Management_Agent Metrics
//
// Metrics
//
// Gets the current metrics
//
// ---
// produces:
// - application/json
//
// Schemes:
//   - http
//
// Responses:
```

edgexfoundry.org | @edgexfoundry

EDGE X FOUNDRY™

Meeting Minutes

Approved

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

Approved

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

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