AI-powered risk assessment before merging code: Developers should get a "risk score" for each PR before merging.

- Can do with our PRs and support data
- Methodology
    - Support data should include
        - PR to fix the support issue
        - Any data that might help us decide on the complexity of the issue - e.g., time took, number of comments, number of people involved - TODO we need to evolve this
    - For each support issue, we can get the PR for the fix, and use blame on each line changed to find what PRs could have caused it and a give a risk score for the causing PR
    - Using this data, we can give a risk score for each PR in the system
- ~~Can we pull data from support issues, use blame to find what PRs cause the problem, and give a cost score for the PR based on how much time was reported to fix it~~
    - Then, we build a model to forecast the cost given a PR
- High risk PRs, we can do code reviews, more reviews etc

Repos and issues

- TODO
- This is what we used to create issues up to APIM and MI product mearge(until 4.0.0) https://github.com/wso2/product-apim/issues .
- After the merge, we started tracking APIM MI and SI issues in https://github.com/wso2/api-manager this repo.
- Additionally, we managed MGW and Choreo connect related issues in https://github.com/wso2/product-microgateway/issues
-
-

How to create the data set?

- Get all the resolved issues
- Get
    - Title
    - Description
    - Comments
    - How long did we worked on the issues?
    - Patch
    - TODO - we might need more
- Develop a complexity measure for the issue
- Get the patch associated with each issue

    - Use the blame to find commits that last update what was patched
- Assign the complexity measure for each commit identified via blame

How to build the model?

- Map the code in the patch to world embeddings and train a neural network?
- Explore how code can be represents as embeddings
- Explore this more