General

In addition to serving as a check for compliance with guidelines, code reviews can be a good way to promote collaboration and minimize silos. All developers should participate in code reviews. Developers are encouraged to review pull requests from all areas, not just the areas they know best.

Code reviews take time. They are a source of friction against the important goal of continuous integration. But the value gained should justify the cost. Make code reviews a priority.

Code review checklist

The checklist is a guideline. It is not comprehensive. Opinion, experience, and subjective judgement all play a role as developers, together, decide the standard.

Code correctness

- Is the change consistent with naming and style guidelines?
- Is the change covered by unit tests?
- Does the change build without error?
- Do the unit tests pass?
- Is an incomplete feature exposed?

Maintainability

- Did the change introduce duplication?
- Are deprecated code and comments removed?
- Does the change introduce hard-coded values? (magic string, magic number)

Migrations

- Verify that each commit with a migration change includes a label in square brackets for the work item number.
- Verify that transient (not re-runnable) migrations include a timestamp in the file name.
- If the migration updates the data dictionary, does it include a step to update the online tables?

Are there other obvious problems?

Code review ceremony

- When the feature is ready, the developer creates a pull request to merge the feature branch into the mainline branch.

- Two reviewers are required, other than the developer who creates the pull request, to approve each pull request. As a general guideline, one reviewer should be a PSI team member and the other reviewer should be a Nucor or Cap team member.
- The pull request will automatically check for merge conflicts, build*, and test the feature branch code.
- A code review will be required to approve the pull request. The merge to the mainline takes place upon approval. If the feature is complete, the feature branch is removed.

Work tracking tool

As an integral step within the code review process, it is imperative to ensure the accuracy of the information in the work tracking tool. Here are the necessary actions to be taken based on the context of the project being tracked in ADO:

1. If the requirement is part of the project being monitored in ADO:
    - The Pull Request (PR) should be linked to the corresponding Feature.
    - The Feature branch must also be connected to the respective Feature.
2. In cases where the requirement is associated with the project being tracked in ADO but also requires synchronization with another project tracked in ADO (e.g., NGLO):
    - Both the PR and the Feature branch need to be linked to the ADO Feature. It is recommended that the Business Owner or the Product Owner responsible for this requirement perform this linking even before the development phase commences.
    - If, by the time of the Code Review, the Feature does not yet exist, it becomes the developer's responsibility to coordinate its creation. The Feature needs to be established if it is absent.