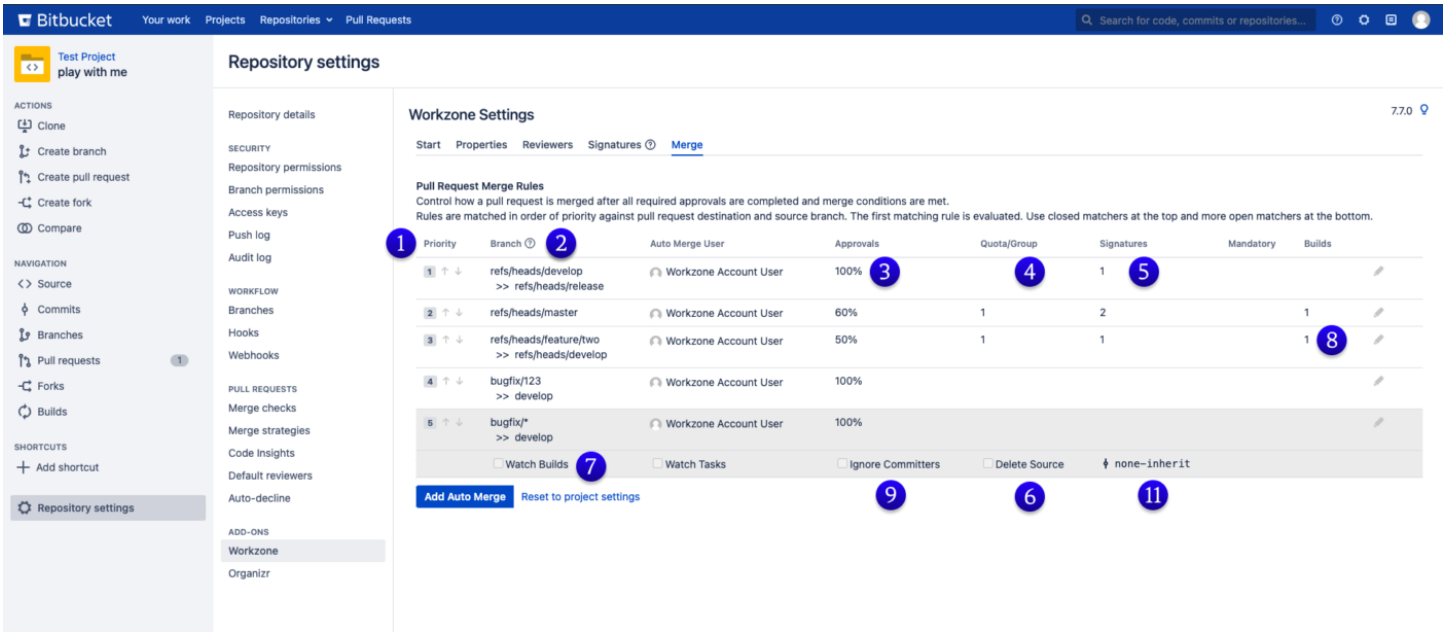
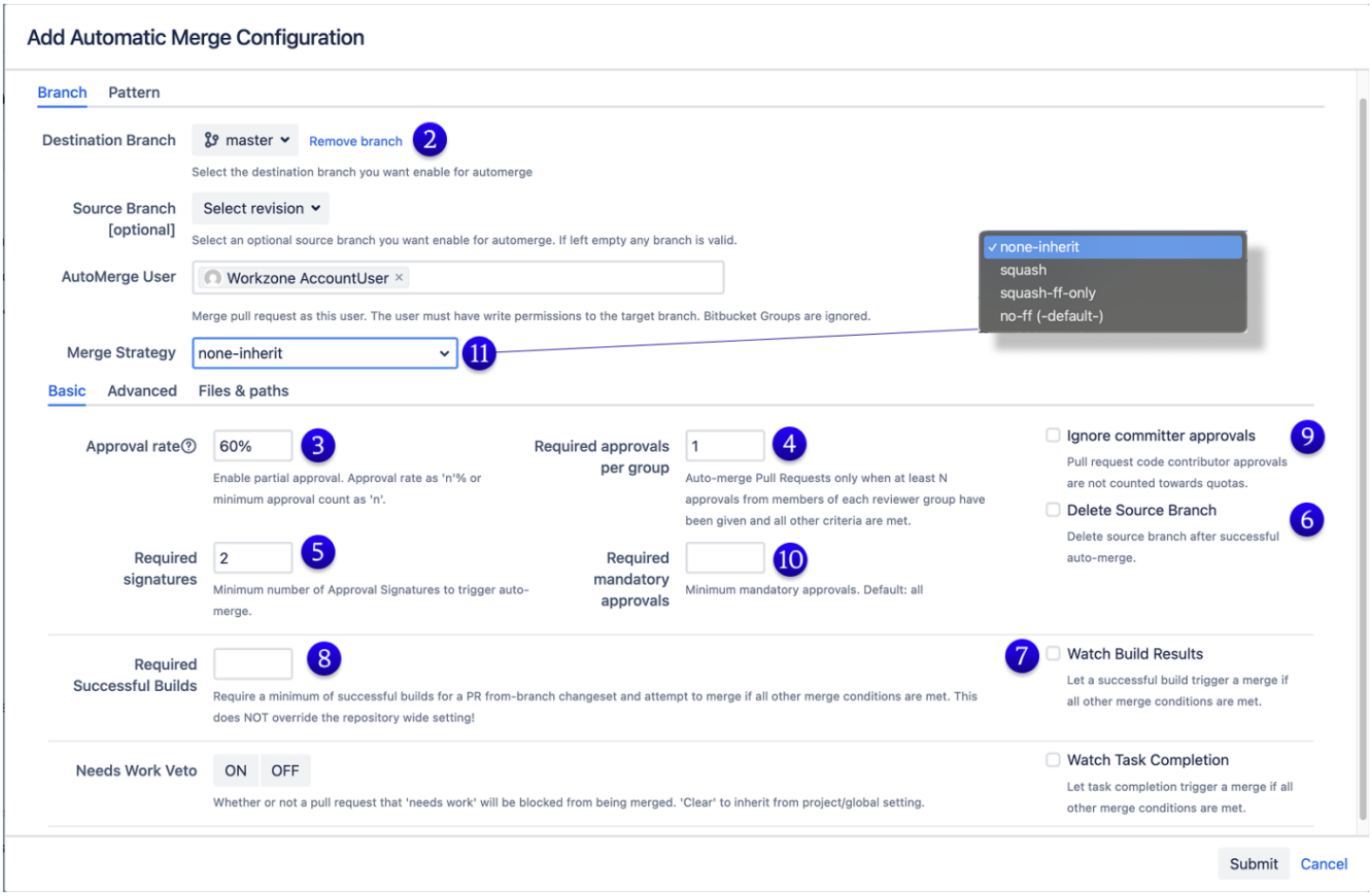
Merge Control Overview: Configure automatic pull request merge

Feature Guide Workzone · https://izymes.com/docs/workzone-merge-control-configure-automatic-pull-request-merge/

Workzone implements fine-grained pull request merge control per branch or branch pattern.

- Workzone can merge the pull request automatically when all or a percentage of reviewers have approved and/or on successful build results.
- Workzone can delete the source branch of a pull request after successful automatic merge.
- Allow or block push operations to a branch once a pull request has been created and un-approve all approvals on a pull request should the originating branch be updated.
- Can block a pull request from being merged manually.





The Workzone auto-merging feature has the following options and are combined to allow a merge to happen or block it:

1. **Priority:** The first rule that matches a pull request's source and destination branch will be executed.
2. **Target Branch:** the target of pull requests that should be merged automatically. Can be a branch name or pattern. (Optional: **Source Branch:** the source of pull requests that should be merged automatically. Can be a branch name or pattern.)
3. **Reviewers approval quota:** how many % of reviewers must approve
4. **Reviewers group quota:** how many members of a reviewer group must approve
5. **Required digital signatures:** how many approvals must have a digital signature for CFR Part 11 compliance.
6. **Delete Source Branch:** when a pull request is automatically merged, delete the source branch of the pull request immediately after.

7. **Watch Build Results:** triggers an automatic merge of a pull request on successful build result for the source branch commit, provided all other conditions are met.
8. **Required builds:** how many successful builds are required before the pull request can be merged.
9. **Ignore committer approvals:** Pull request approvals from reviewers that also committed code to the pull request changeset are ignored when a merge-check is performed.
10. **Required Mandatory Approvals:** How many approvals must be given from mandatory reviewers
11. **Merge Strategy:** Define a Pull Request auto-merge strategy that will be used by Workzone. Workzone can use any merge strategies that are enabled repository or project wide

Auto-merge user

The auto-merge user is the Bitbucket user on which behalf a pull request merge is performed automatically in the background when all merge criteria are met.

Tip: It is best practice to use Bitbucket branch permissions to *limit write access* to the pull request target branches like ‘master’ or ‘develop’ to ensure ‘normal’ users can’t just press the ‘Merge’ button.

Define a user that will perform the automatic pull request merge to the target branch once all approvers have given their approval. This user must have write permissions to the target branch, in fact this user should not be a real-life person (see here for more detail). The shadow-like auto-merge user should be the **only** one allowed to commit to the target branch (usually master) with the correct write permissions for branch, repository and project. You may also give write permissions to a ‘normal’ you can trust in case a pull request needs to be merged manually via the ‘Merge’ button.

No auto-merge required

If you don’t want Workzone to trigger an automatic pull request merge but still want to control the manual merge process, **leave the auto-merge user field empty**. Workzone won’t merge pull requests to the specified target branch automatically. You can still enforce merge checks by enabling "enforced merge conditions" for the repository or project.

Try Workzone!