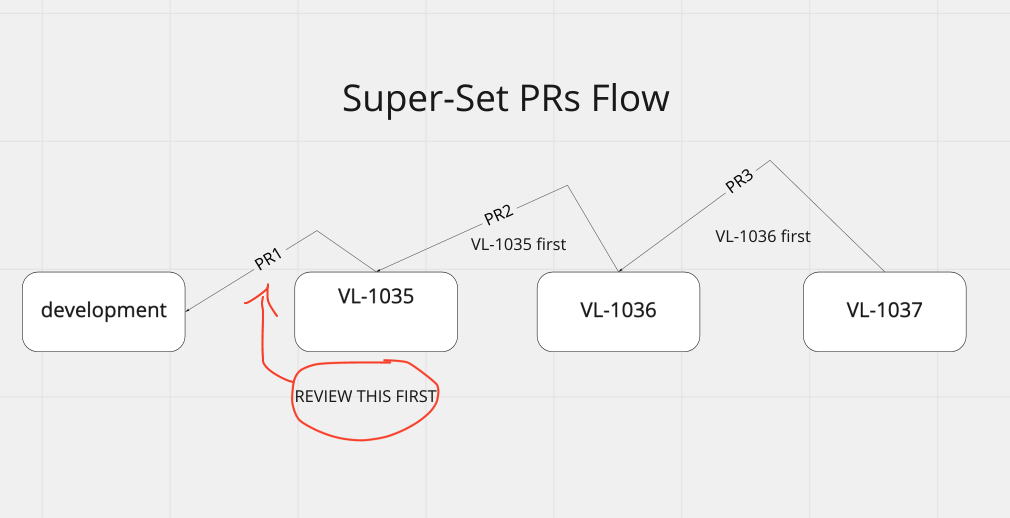
Guidelines for creation of PRs

1. When creating a new branch we always want to name it with the ticket number from RR plus part of the ticket’s title ( i.e. VL-1038-generic-dialog-component.
2. PR’s shouldn’t have changes for more than 15 files per time.   
   If the developer evaluates that more changes could occur, then it has to be broken down into smaller tasks and follow the “reverse merging strategy”.
3. **Reverse merging strategy:**



Pic - Super-Set PRs Flow

When the code that a developer is writing doesn’t have any dependency from another branch, the code should branch out from “development” ( or any other base branch ) and then the subsequent PR should target again the development branch.  
This one can be identified as the “normal flow”.

When the developers recognize that subsequent changes might be a super-set of the code already developed on another branch, they should diverge from that branch. The pull request that will follow, needs to target the parent branch, as shown in the picture “Super-Set PRs Flow”.  
The developer should also highlight in next PRs titles or descriptions ( PR2 and PR3 in the pic ) that the previous PR should be reviewed first ( VL-1035 first, VL-1036 first etc. ).  
When in code review phase PR1 is accepted and merged, BB automatically re-targets PR2 to development.

Benefits of using this model are:

* Better readability in PR ( the PR shows only the relevant delta )
* Easier code review
* Less problems in case of reverts
* Teams can start to merge previous PRs even if subsequent others have bugs to fixs or tweaks to address.