1 . Version Control

The latest version of vue storefront is 1.12.3. They added states.json in core/i18n/resource , Added phone validation helper , Configurable enabling min & max price aggregations , Storing totals in localStorage to sync it between tabs and Support for trailing slashes in the route paths . Some issues are also resolved in latest version 1.12.3 such as Fix gallery image generation by checking if image exists , Fix getSelectedOption based on attribute_code check , add new resetUserInvalidation action to clear invalidation state before login, clear order history and refresh token after logout, Multi-tab cart-sync in multi-store environment , Incorrect load of default address in checkout and Fix Order History Pagination.
Moreover , changes and improvements are made in the latest versions such as

- Moved hard coded fields from omitSelectedVariantFields.ts to config #4679.
- Bump dependencies versions such as #4715,#4696and #4951.
- Using days for dates in taxCalc.ts to make it work properly in Safari #5364.
- Awaiting addItem action call inside mergeServerItem action.
- Moved Phone Num to proper branch.
- Development hot-reload speed webpack config - #5559

2. Release Management Control

Vue storefront has 50+ releases and the latest version is 1.12.3. Vue storefront has 274+ contributors and languages used in these releases are TypeScript , Vue.js , javaScript , HTML and have a DockerFile.

How Vue Storefront versions are released

From version 1.9, vue storefront release each of the VSF versions in two phases:

- Release Candidate phase (RC), also called feature version. This version contains all the new features, improvements, and additions to the API, along with minor bug fixes. New features and additions are merged and released only during this phase. The API of features introduced during this phase may slightly change.

- Stabilization phase is the one that ends up with a production-ready version. During this phase, they only do stabilization and bug fixing for previously introduced features. No new features and API additions are merged. PRs from the RC version are tested and their API is simplified and/or adjusted according to feedback.

Assuming the next version is 1.x, the two-month cycle will look like the following:

- v1.x-RC.y—unstable version with cutting-edge features ready to test and feedback.
- v1.x.y—stable version of the software ready for production use.

● How new features are merged

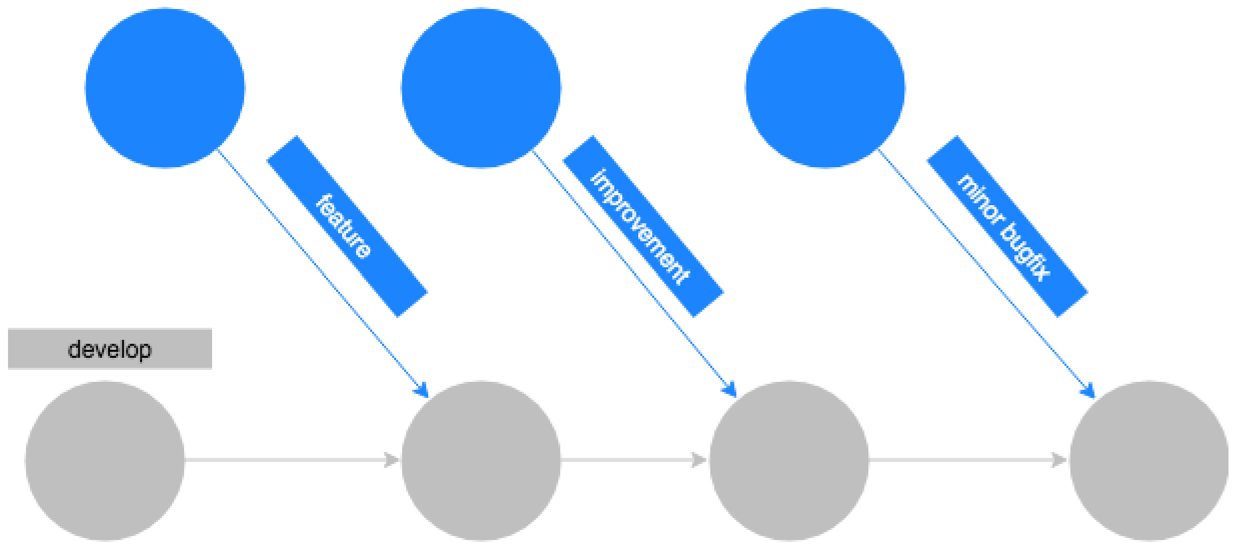
During the RC phase, features Pull Request with new features after feedback and acceptance are normally merged to the Develop branch. After entering the stabilization phase, they are tagging the current develop branch, creating a release/x (where x is the number of the current version) branch from it and working on stabilization there. During the stabilization phase, new features are merged to develop the branch and will be merged in the next RC phase.

● Release cycle flow

- **Development Phase**

In the first phase of the cycle, they're mostly focusing on features and improvements. Branches in this phase should be created from the actual development, also PRs should be pointed to this branch. Changes merged to develop are available to test on.

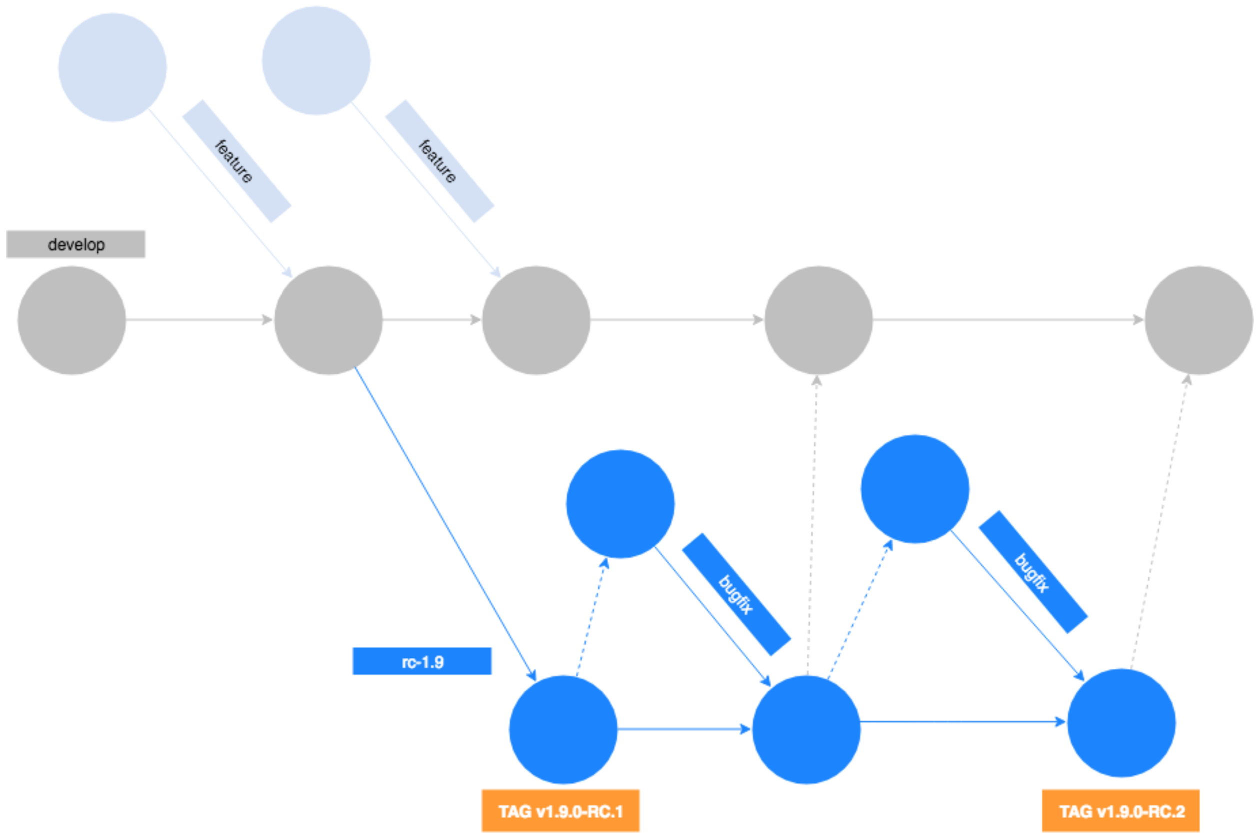
Here is the pictorial representation



- **Release Candidate phase**

At some point, when the milestone for next minor versions is completed, they are creating a new branch from the development called release/vx.y (example: release/v1.9). After that new branch is tagged as the first RC for version (example v1.10.0-rc.1). Then it's ready for testing by the community. During tests, feedback and stabilization there could be multiple Release Candidate versions on this branch. When improvement is made on this phase, then branches should be created from actual release/vx.y and should not contain features at this point - only improvements for current release. After merging a set of bug fixes and improvements into the release branch, it needs to be tagged as the next RC version and merged into the develop branch, to update it.

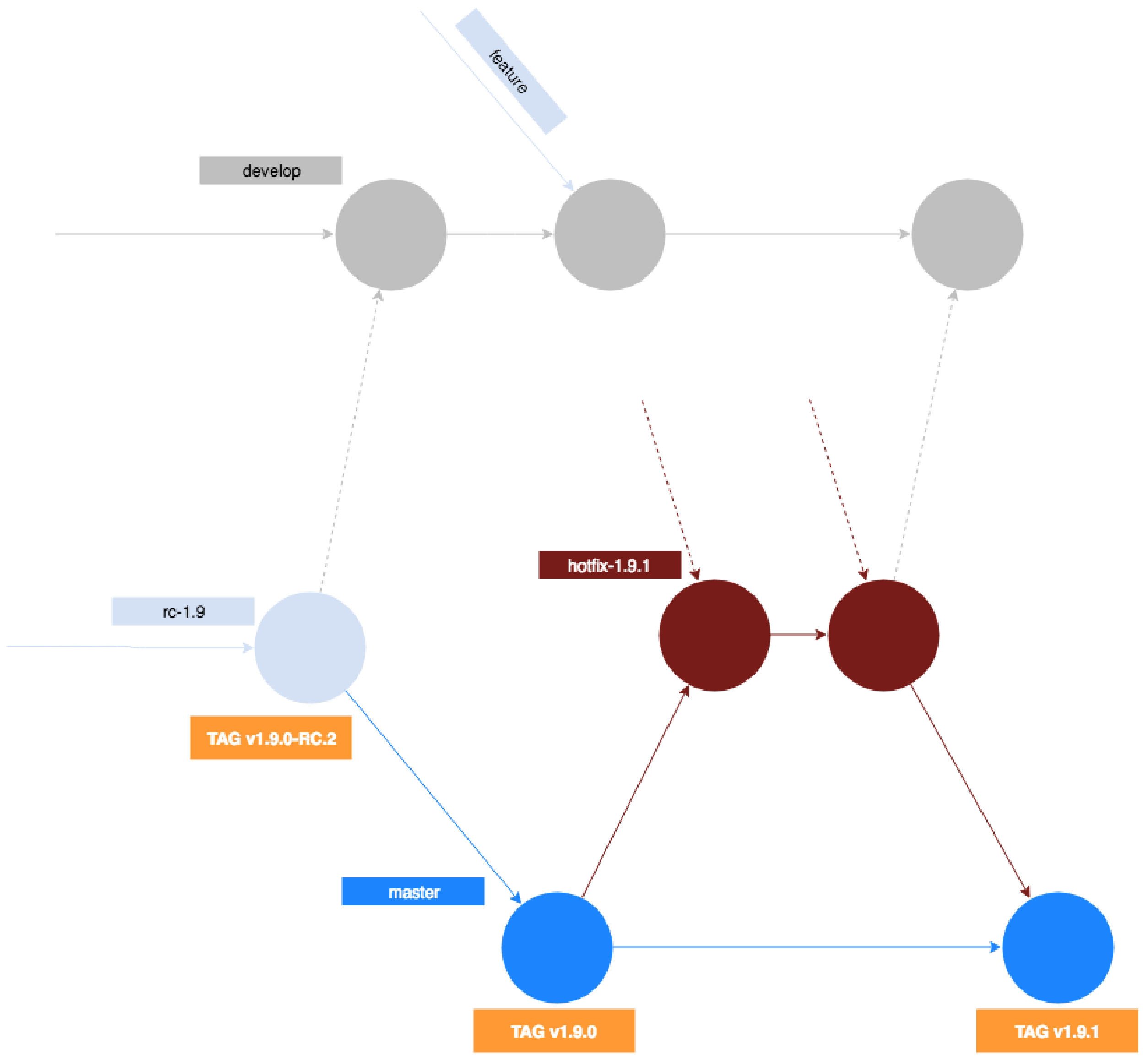
Here is the pictorial representation



- **Release phase**

When the RC version is stable, the release branch is merged into master and tagged as the next stable version (example v1.9.0). After that, the currently merged release branch is deleted and starts a new development phase. If some critical bug is discovered on the current stable version,) branch should be created from the actual master. After merging and testing hotfixes, this branch (like the release branch before) is merged into master.

Here is the pictorial representation.



- **Summary**

An important thing to note is that this releasing cycle ensures stable and reliable Storefront releases. Phases are also not blocking new features—you can develop new features on any phase, but it should be merged only to the develop branch and go through the whole cycle.

3 . Open source contribution requirements for developers

If you are a JavaScript/Vue.js developer then Pick an issue and push a pull request (PR) and instantly become a member of the vue-storefront contributors community.

You can start a ready-to-code development environment in your browser.

Before you type an issue please read about their release lifecycle.

- Branches

You should fork the project or create a branch for new features. The main branches used by the core team are:

- master - where they store the stable release of the app (that can be deployed to their demo instances),
- develop the most recent version of the app - kind of nightly build.
- RC-x (x is the current version) - release candidate branch with features that will land in the next version.

Please use develop or RC for development purposes as the master can be merged just as the new release is coming out (about once a month).

- **Issue Reporting Guidelines**

Always define the type of issue:

- Bug report
- Feature request

While writing issues, be as specific as possible. All requests regarding support with implementation or application setup should be sent to this email

contributors@vuestorefront.io.

Tag your issues properly. If you found a bug, tag it with a bug label. If you're requesting a new feature, tag it with the feature request label.

- ## Pull Request Checklist

Always use the Pull Request template, it is automatically added to each PR.

1. Fork the repository and clone it locally from the develop branch. Make sure it's up to date with current develop branch
2. Create a branch for your edits. Use the following branch naming conventions:
- bugfix/task-title
- feature/task-name
3. Use the Pull Request template and fill as many fields as possible to describe your solution.
4. Reference any relevant issues or supporting documentation in your PR (ex. “Issue: 39. Issue title.”).
5. If you are adding new features, provide documentation along with the PR. Also, add it to upgrade notes. Below is the link of upgraded notes.

   upgrade notes

6. If you are removing/renaming something or changing its behavior also include it in upgraded notes. Below is the link of upgraded notes.

   upgrade notes

7. Test your changes .Run your changes against any existing tests and create new ones when needed. Make sure your changes don’t break the existing project. Make sure that your branch is passing Travis CI build.
8. If you have found a potential security vulnerability, don't report it on the public issue tracker.
9. Instead, send it to this email :

   contributors@vuestorefront.io.

- **Acceptance Criteria**

Your pull request will be merged after meeting following criteria:

- Everything from "Pull Request Checklist"
- PR is proposed to appropriate branch
- There are at least two approvals from core team members

4. REVIEW PROCESS

If any developer wants to contribute in vue store front then the developer has to report that issue and vue storefront team do their best review in a reasonable time.

5 . ISSUE LIFECYCLE MANAGEMENT

Following are the guidelines when reporting issues:

- Provide a title in the format of Ex:[BUG] : <Error> when <Task> , [Issue] : When i try to <x> , an <Error> appears.
- Tag your issue with the tag triage-needed.
- Provide a short summary of what you are trying to do
- Provide the log of the encountered error if applicable
- Provide the exact version of the framework you are using.
- Be awesome and consider contributing a pull request

6. INSTALLATION

Following is the installation link of document of vue storefront:

https://docs.vuestorefront.io/v2/getting-started/installation.html

7. TUTORIALS

Here is the link to the official youtube channel of vue storefront.

https://www.youtube.com/c/VueStorefront

8. USER MANUAL

Here is the link to the user manual of vue storefront.

https://vuestorefront.io/blog/quick-vue-storefront-getting-started-guide