Behat is an open source, community-driven project. We would love you to contribute to

this awesome project, but remember to follow the few simple rules defined in this

document.

## Submitting an Issue

1. State clearly if it is a feature, a problem or refactoring. You can even use one

of the [GitHub labels](https://github.com/Behat/Behat/labels) we carefully created

just for that purpose. That makes our life easier as we tend to focus on resolving

issues quicker than adding new features

2. Outline clearly in one/two sentences why that feature is important to you or why

that problem causes you grief (and at what scale). This helps us properly prioritise

what we're working on

3. Behat is [automatically tested](https://travis-ci.org/Behat/Behat) on every change.

If you have a problem, chances are high it is something very specific to your context

and the more we know about it the more likely we would be able help. At the very least

provide us with enough information about your feature files, context classes and local

environment

4. Make sure you stay professional and do not use offensive language in your issue.

Positive tone in the text of issues is known to reduce lead time to help

5. If [asked for clarification](https://github.com/Behat/Behat/labels/requires%20clarification),

we expect to hear back from you in 7 days. If no answer is given in 7 days, issue will

be automatically closed. You can easily open new issue again later and this rule helps

us reduce the clutter of "silent issues"

## Submitting a Pull Request

1. Make your feature addition, bug fix or refactoring

2. Add new `Scenario` into [one of existing features](features) or create a new `.feature` file describing

the changes. Check the [features/](features) folder for examples. This is important so we don't break the

changes you introduced in a future version unintentionally

3. Make sure your changes adhere to [Backwards Compatibility](#backwards-compatibility) rules. This is important

so that we adhere to [Semantic Versioning v2.0.0](http://semver.org/spec/v2.0.0.html)

4. Explain the kind of change you made under the [`[Unreleased]`](CHANGELOG.md#unreleased) section of the

[CHANGELOG.md](CHANGELOG.md). You'd make our life even easier if you stick to [Keep a Changelog](http://keepachangelog.com/en/0.3.0/) format

5. Do not mess with the [`BehatApplication::VERSION`](src/Behat/Behat/ApplicationFactory.php#L48) version

6. Make sure you [ran tests](#running-tests) and didn't break anything. That will save some time for

[Travis](https://travis-ci.org)

7. Commit your code and submit a Pull Request, providing a clear description of a change,

similar to the one you did in the changelog

## Backwards compatibility

Starting from `v3.0.0`, Behat is following [Semantic Versioning v2.0.0](http://semver.org/spec/v2.0.0.html).

This means that we take backwards compatibility of public API very seriously. So unless you want your PR to start a

new major version of Behat (`v4.0.0` for example), you need to make sure that either you do not change existing

interfaces and their usage across the system or that you at least introduce backwards compatibility layer together with

your change. Not following these rules will cause a rejection of your PR. Exception could be an extremely rare case

where BC break is introduced as a measure to fix a serious issue.

You can read detailed guidance on what BC means in [Symfony2 BC guide](http://symfony.com/doc/current/contributing/code/bc.html).

## Running tests

Make sure that you don't break anything with your changes by running the test

suite with your locale set to english:

```bash

$> LANG=C bin/behat --format=progress

```