

# Meetings

Thursday, November 02, 2017 6:55 PM

# Shiva

Thursday, November 02, 2017 6:56 PM

# Code conventions in Ruby

Thursday, November 02, 2017 6:56 PM

- Changes to existing project structure to enable documentation generation.
- Using classes concept to leverage benefit of OOPs concepts. This will be helpful while implementing BDD.
- Customize XPATHs to an extent that it minimize the extensive coding.
  - Example, to retrieve cell value from a table. Refer to customized XPATH written by Krishnateja for getting values from GroupID.html.
- Minimize XPATH inventory.
- Avoid redundant verification of similar feature in multiple test cases.

# Ruby project structure for GIT

Tuesday, December 05, 2017 4:46 PM

## Suggestions:

1. In **fds-qa-test-common.rb**, use relative paths.
2. Use small case letter for Test Plans and Test Cases.
3. Maintain feature related scripts in single folder. For example, if a particular feature has 20+ test cases which is split b/w 2-3 test plans on Justifier; On automation side one should still maintain all scripts related to testing that feature in single folder.
4. Start using rspec format to develop test scripts in Selenium-Ruby - POC is required.
5. Use test-config.json to get the list of test cases to be executed. Tellus code should not be responsible for dynamically writing this information.
6. Can we come to common name for **qa-protractor-common** & **fds-qa-test-common**?  
**fds-qa-test-common** should probably be changed to **fds-qa-ruby-common**?

# Tricentis Tosca Evaluation

Friday, January 05, 2018 10:45 AM

## Software Requirement Review:

- Need high system resources. Ex.: 4GB RAM, i5 Processor
- To work on browser Tosca extension has to be installed specific to browser.
- Uses native browser installed on machine rather than using drivers.

## Feature Review

- Intuitive UI, easy to use.
- Support testing of APIs, GUI, non-GUI and many.
- No Coding is required thus non-technical person can use it.
- Support cross browser but only on Windows OS.
- Good customer support.

## Pros:

- Good Documentation available but in order to use the tool lot of reading is required. But, once you get used to options, you can automate test cases quickly.
- Lot of standard modules available to accomplish basic tasks (but most of them is missing in Trial version).
- Can create reusable steps and distribute it for automation of other applications.

## Cons:

- As we have web-components build with custom libraries like thief-angular, thief-grid etc., standard methods finds it difficult to steer the application. For example, with standard html component for dropdown it can identify underlying options but for the drop-down developed using thief library it doesn't identify options automatically.
- Many times it fails to identify elements during execution if application takes a little more time to load and thus fails execution.
- Takes away system control while execution in progress.

## Features to Verify

- Ability to identify dynamic elements. Example, typeahead elements, injecting headers.
- Cross-browser testing.
- Using XPATHs to identify elements.

**NOTE:** This is my preliminary review. As I work more with the tool I'll be able to provide further details.