

## Best Practices

1. Make sure your code is indented properly. Use Sonarlint plugin if you need automation
2. Go through the framework, regression tests and all its methods in detail.
3. If you want to add wait then please don't add `Thread.sleep()` method, instead you need to add like new `WebDriverWait`  
(`WebDriverAccess.getDriver()`, `<timeInSeconds >`).until(`ExpectedConditions.visibilityOfElementLocated`  
(`By.xpath("//div[@id='ui-datepicker-div']")`));
4. Always use `CommonPage` map to use parameters for 'timeInSeconds'. e.g. If you want to pass waitTime use `CommonPage`.  
`getTestData("WaitTime")`.
5. Do not use try-catch blocks, until you want to perform any action due to that exception.
6. Always use custom exceptions whenever and expected condition fails.
7. Use data from csv file only.
8. Use camel case for the method name and variable names with small character as its first letter.
9. Enums should be in capital case separated with underscore ("\_") between words..
10. Give meaningful name for methods, variables, xpaths and enums. Also limit the length of the names for the same(methods/variables).  
Don't give huge name and it should be readable. (Method name should not have more than 30-35 chars).
11. Add the comment block at the start of the test method defining the purpose say

```
/**
 *This method perform Addition of New Approval Rule.
 *@lastModifiedBy unaidu
 *@param option
 *@throws Exception
 */
```

12. Use relative path and not the absolute path. If possible always try to find the locators by id. The reason behind it that even if the UI layout changes, our existing code will still work because we are finding the element by its id.
13. Remove unnecessary classes/packages from the imports. Remove the extra blank spaces throughout the class which you are modifying.
14. Execute the code once script completed.
15. Run the same code in Jenkins under job - Test Job and save the build ID which will be used later.
16. If you want to create new method/function then always try to build the code in a generic way so that it will be re-used by others. This practice will restrict the code redundancy. Our application have pages which have similar UI layout. So to perform any action we can build the code in a way where others can re-use the same code e.g. like in BO there are queues like Billing Queue and Dealer Queue which have similar UI layout. So we have created methods which will work for both queues. Please refer theses classes :  
`Reg_BillingQueue_Search_Sort_ToggleColumn_ClearFilter_Export /`  
`Reg_DealerQueue_Search_Sort_ToggleColumn_ClearFilter_Export.`
17. If you are modifying existing method/function for your functional requirement, then check that method's references throughout the existing implemented scripts. Run all the scripts where that methods reference is e.g. if method "M1" is been modified by you for your script "S1" and Method "M1" is also has reference in Script "S2", "S3". Then after your modification of method "M1" run the script "S1", "S2" and "S3" as these Scripts has the impact of method "M1" modification.
18. Use relative path and not the absolute path.

Ex. `System.getProperty("user.home")+"\\Downloads\\"`

19. If possible always try to find the locators by id. The reason behind it that even if the UI layout changes, our existing code will still work because we are finding the element by its id.
20. Don't use wildcard locators ("\*") in xpaths.
21. Remove unnecessary and deprecated classes from the imports. Remove the extra blank spaces throughout the class which you are modifying.
22. Don't use static imports

Ex. `import static org.testng.Assert.assertEquals;`

23. Execute the code once script completed.
24. Run the same code in Jenkins under job - Test Job and save the build ID which will be used later.
25. Ordering Exception should be thrown for error/exception
  - `throw new OrderingErrorOccured("Add Comment");`
  - `throw new NoIfElseBlockMatchedException("Add Comment");`
  - `throw new InvalidSwitchCaseException("Add Comment");`

26. Common method should be used if available or try to create common method and keep in CommonPage.java page.
27. Peer review should be done before raising PR.
28. Avoid global variable declaration and if declared use private as its access- modifier.