In industry-level frameworks, the best practice is to handle SoftAssert at the BaseTest level for several reasons:

1. **Centralized Assertion Management**:

* All test classes inherit the same assertion handling
* Consistent assertion behavior across all tests
* Easier to maintain and modify assertion logic

1. **Better Test Class Organization**:

* Test classes focus on test logic and scenarios
* Base class handles common functionality like assertions
* Follows Single Responsibility Principle

1. **Reduced Code Duplication**:

* No need to create SoftAssert in each test method
* Common assertion patterns can be added to base class

This implementation follows industry best practices because:

1. **Centralized Assertion Management**:

* SoftAssert is managed in BaseTest
* Thread-safe implementation using ThreadLocal
* Clean setup and teardown in @BeforeMethod and @AfterMethod

1. **Clean Test Classes**:

* Test classes focus on test scenarios
* No need to manage SoftAssert lifecycle
* Consistent assertion pattern across all tests

1. **Additional Benefits**:

* Easy to add common assertion methods in BaseTest
* Can add custom assertion logging
* Can implement retry mechanisms for flaky tests
* Can add custom assertion failure handling

1. **Industry Standard Features**:

* Thread-safe for parallel execution
* Clean separation of concerns
* Easy to extend and maintain
* Follows Page Object Model principles

Would you like me to add any additional assertion-related features to the BaseTest class?