



BLOCK NONCE #13: Informative Test Error Message

What?

When a test fails, the error message should provide enough information to help understand the issue.

Why?

Help future developers debug and maintain the code.

The following code prints a non informative error message.

```
#[test]
fn my_test() {
    let result = TransactionExecutor().add_tx(reverted_tx());
    assert!(result.is_ok());
}
```

```
thread 'my_tests::my_test' panicked at my_tests.rs:37:5:
assertion failed: result.is_ok()
```

Provide an error message argument to the `assert!` macro. Print what matters.

```
#[test]
fn my_readable_test() {
    let result = TransactionExecutor().add_tx(reverted_tx());
    assert!(result.is_ok(), "Transaction execution failed: {result:?}");
}
```

```
thread 'my_tests::my_readable_test' panicked at my_tests.rs:43:5:
Transaction execution failed: Err(Reverted("impl detail"))
```

The following code uses `assert_matches!` which provides clear error messages when the assertion fails.

```
#[test]
fn my_other_test() {
    let result = TransactionExecutor().add_tx(test_tx());
    if let Err(TxExecutorError::Reverted(message)) = result {
        assert!(message.contains("impl detail"));
    } else {
        panic!("Expected Reverted error, got: {:?}", result);
    }
}
```

```
thread 'my_tests::my_other_test' panicked at my_tests.rs:49:5:
Expected Reverted error, got: Ok(())
```

Use `assert_matches!` for pattern matching assertions. It provides clearer error messages.

```
#[test]
fn my_other_readable_test() {
    let result = TransactionExecutor().add_tx(test_tx());
    assert_matches!(
        result, Err(TxExecutorError::Reverted(message))
        if message.contains("impl detail")
    );
}
```

```
thread 'my_tests::my_other_readable_test' panicked my_tests.rs:58:5:
assertion failed: `Ok()` does not match
`Err(TxExecutorError::Reverted(message)) if message.contains("impl detail")`
```

TIP 💡 When writing the test, insert a bug and see how the error message prints.

Call For Action! 📣 When maintaining a test and you encounter a non informative error message, provide one yourself.

 #starkcode