# Fail early, fail loud!

The "require" statements at the beginning of almost every method in my smart contracts will make sure that the user inputs are valid before continuing the execution. If a failure occurs, the contract function exits immediately and can be caught on the UI if required.

# Restricting Access

I also used the "require" statements to check the msg.sender and verify that they are indeed allowed to execute certain portions of the code. If they are not, it would fail.

# Circuit Breaker

The emergency stop procedure that can be called by an admin, just in case something goes out of control and needs to stop. Important methods have a "stopped" modifier to carry this out.

# EthPM

I manually imported Ownerable.sol into Marketplace.sol, So system designed in such way may enable upgradability.

Notes: I had error message in using ethPM truffle install zeppelin. Instructor Josh told me to manually import the contract.