I am learning how to develop Secret Contracts, and I am looking at some example implementations for SNIP-20 tokens etc. I came across something that seems like a bug in the reference, but I wanted to make sure my understanding is correct before jumping to conclusions.

Referencing the try_deposit method, for isntance [snip20-reference-impl/contract.rs at master · enigmampc/snip20-reference-impl · GitHub]

There are several error checks while depositing uscrt native tokens to get sscrt tokens minted. One check makes sure the deposit wouldn't overflow the balance. The balance is then set if the check passes. The next check is that the deposit wouldn't overflow the token supply limit. Supposing this failed, wouldn't our Smart Contract method have left the contract in a bad state where a balance has tokens that are not accounted for in the total supply?

The fix for this is to put all the checks at the beginning of the method and only call the state setters when all checks pass. Am I correct in this assumption? Or is there some feature of cosmwasm storage where all changes are only committed if the Secret Contract method returns OK and doesn't err?