Guidelines for TUCBW/TUCEW descriptions in your EUCs - **these go in the Exp UCs for It 2 only. Note that this information comes from your UCID tables.**

1. Do the TUCBW and TUCEW have some meaningful description?
2. Do the TUCBW and TUCEW avoid using implementation level terms that could change later such as "click", "button", "menu", etc. Page and screen **are acceptable** because the UCID will use that term.
3. Does each TUCBW start with a user action? Does it state where it came from?
4. Does each TUCEW end with a user action? Typically this is going to be a confirmation that the user sees something happening. Does it state the redirection if the UCID shows one? Does it also talk about the navigations allowed?

Step 1 - Expanded Use Cases (Exp UC)

1. Are there pre-conditions and post-conditions for each?
   1. Are they non-trivial - are they something besides "None" and describe something relevant?
   2. Does the post-condition describe the system operation - what the system promises to provide? Each should start with "They system..."
2. Are there TUCBW/TUCEW for each? See above guidelines.
   1. Do the TUCBWs describe what action initiates the UC?
   2. Does the TUCEW end with a positive action or confirmation by the actor?
   3. Does the TUCBW/TUCEW avoid using click, buttons (screens are okay)?
3. Steps
   1. Does it start at Step 0 (system step)?
   2. Do it end on an actor step (odd number step)?
   3. Is each step numbered sequentially?
   4. Are there no empty steps in the use case interactions?
   5. Does each actor step clearly and correctly specify the actor input and actions?
   6. Does each system step clearly and correctly specify the corresponding system responses?
   7. Are non-trivial steps (on the System side) indicated?
4. Reference to UIPs
   1. Is there a clear reference from the System steps to each UI prototype figure number reference?
5. Non-trivial steps identified? The EUC should show which steps are non-trivial
6. Consistency with other artifacts
   1. Do the delivered Exp UCs match what is in the Increment Matrix?

Step 2 - UI Prototypes (UIPs)

1. Do attributes listed on each UIP correspond with attributes in the domain model? With the DCD? With the UCID?
2. Each System side step in the EUC must be shown as a UIP
   1. Are example UI prototypes provided non-trivial - do they provide the user with representative data - do they avoid listing things like "Item 1", "Item 2", etc and instead provide actual names?
   2. Do buttons mentioned on internal steps (not TUCBW/TUCEW) refer to the same buttons as the UIP. Typical mistakes are Register button (EUC) vs. Submit button (UIP) for the same EUC step.

Step 3 - Sequence diagram (SD)

1. Have 2 SDs been delivered? Is only one of them login, logout, register, update profile?
2. Is each non-trivial step in the expanded UC captured as a SD? Do steps from the expanded use case match the flow on the SD?
3. Does each actor have the same name on the SD and UCD? Is it the same as the Exp UC?
4. Is each message formally specified? Are return values specified as "return\_value:=..." instead of as dashed arrows back to the calling object?
5. Does the flow go from top to bottom and left to right? Can a return type be traced from where it is sent from the user to back where the data originated? Are all parameters and return values specified?
6. Are lifelines depicted correctly? Are method execution bars depicted correctly on both ends of the reference?
7. Are all actor/object interactions depicted correctly? Are all object/actor interactions depicted correctly? Does each message to/from an actor correctly labeled with Stereotypes and dashed lines used?
8. Are only closed ended arrows used? is correct, is not.
9. Are there no cases of poor coupling on the SD - such as control coupling?
10. Have all design patterns been utilized on the SD?
    1. Has the controller pattern been applied (GUI, Controller, DBMgr)?
    2. Has the DBMgr object been labeled with <<singleton>>?
    3. Has the creator pattern been applied as to who should create an object?
    4. Has the expert pattern been applied to show the best provider for the requested data?

Step 4 - DCD

1. Are singletons correctly labeled? DBMgr should always be labelled <<singleton>>.
2. Are all SD relationships correctly depicted? The DCD should be a summary of all of the information from the set of delivered SDs:
   1. From the SD are relationships from classes as parameters, parameters or return types captured as <<use>>?
   2. From the SD are cases where two objects as parameters - depicted as an association relationship on the Domain diagram?
   3. From the SD are all messages to methods in the SD captured as <<call>> relationships?
   4. Do the class names between the SD and DCD agree?
   5. Are there any objects on the SD that do not appear on the DCD?
3. Are all Domain diagram relationships correctly depicted?
   1. Are all aggregation relationships on the Domain diagram identified as <<create>> relationships on the DCD?
   2. Are all Inheritance and/or aggregation from domain diagram captured on the DCD?
   3. Are they correctly depicted as dashed lines with solid arrows?
   4. Do all attributes in the Domain diagram appear on the DCD?
   5. Do all of the classes on the Domain diagram appear on the DCD?