**Attestation How it fits in with the current Risk and Control management system.**

Attestation is a process that will be managed by the SPOCs to capture user responses where they certify their controls for the identified risks in their Risk and Control Library. To initiate this process, there are several steps that have to be taken to ensure that the correct information is flowing from the RCL to the database so that the Attestation tables can be accurate as they are produced from user input.

**Foundations of a successful Attestation Process:**

1. HR data from the automated daily drop has to be manually uploaded to the SQL Server ERMDB database EMP\_MGR tables for both **EMP\_MGR** and **EMP\_MGR\_AttTest**. To get the data uploaded, there are instructions and SQL scripts to change the formatting once it is uploaded which can be found at…. **S:\Enterprise Risk Assessment\AttestationApplication** on the **finadm** shared drive. Use **EMP\_MGR**, then **Trim Names** to get the information in the proper format.
2. Using the correct employee information, the RCL needs to be updated to ensure that there is a control delegate for each control. This can be the control owner, but they must also be specified as the delegate. The RCL delegates should be cross checked with the most up to date employee table to be sure there are current employees assigned to all controls.
3. A recursion must be run to build the hierarchical information of distinct delegates from the RCL, who their manager is, who the manager’s manager is, all the way up to the CEO. This recursion is in…. **S:\Enterprise Risk Assessment\AttestationApplication\Attestation\SQL** and is the “TreeBuild6\_M0\_Version”.
4. The results from that recursion need to be manually copied and pasted into the Attestation book on the data tab in cell J3 for managers and K3 for their reports. This is CRUCIAL to the book working properly. This recursion could be adjusted to create a new table in the ERMDB that could be referenced from the book. That table could be deleted after completion and recreated for each quarter.
5. With the new recursion information added, the blanked out starter book should be set to this version with the new recursion. Then when the SPOC opens it and sends out the 1st tier initiation email, the latest information is included and nothing is defunct.
6. If any of these steps are not completed, there is a huge potential for a mismatch of information as the attestation process is performed and responses are captured.

**How the Attestation Book is operated:**

The attestation book is a piece made of many sub functions that is, for the most part, automated and controlled by user input. There are 6 different experiences that the book can produce depending on who the user is and what role they are playing in the process.

1. **Control Owner / Delegate** – Receives the book in an email from the SPOC team asking them to fill out their responses and submit them for SPOC review.
2. **Manager of Owner / Delegate** – Receives the book in an email from the SPOC team asking them to review their reports’ responses and then fill out and submit their own responses. This type of user will not be able to answer their questions until all of their reports have submitted their own, and the SPOCs have submitted those answers to the ERMDB database.
3. **SPOCs** – The SPOC will send out the books using some of the methods described further down, but they will also receive the responses and manually submit them to the database. They will be able to reset users, send reminder emails, and manage the general flow of information through the book.
4. **Finance** – As users respond to their questions, some of them will be leaving comments about their yes or no answers. Any records that are submitted with comments will be pulled onto the finance tab for review by finance. Finance will determine if the comment is for SOX or not and if is reportable or not.
5. **Developers** – When the cats need wrangling, the Devs will be there as pillars of reason in a world of chaos. They’ll responsible for making sure the databases are in line and referenced correctly. The book will open all tabs and give access to all pieces. Their credentials are managed on the data tab. They have access to testing functionality by default where they can mirror credentials of users.
6. **Unauthorized** – If someone does not fall into one of the above categories, the book will lock down and drop them on a lockout page instructing them to contact their SPOC. For someone to receive the book would mean they are involved, so the only time someone should see this is if they find a copy of the book somewhere that was not directly sent to them.

**The Cogs of the Machine**

The number of modules and amount of code can seem a little overwhelming. Don’t worry. It’s not all that complicated, just horribly optimized. Let’s start with the beginning of the attestation process, who the user is, what they’ll do, and what code will run to accomplish that action. Everything runs automatically except for user input. This is accomplished using button pushes on all tabs.

1. The long process that occurs when the book opens.
   1. Under the Microsoft Excel Objects in the VBA editor there is a component called ThisWorkbook. Click that to see what happens when the book is opened.
   2. First all tabs are defined using Dim statements, then those values are set using the sheets command for reference.
   3. All sheets are made visible in case they aren’t. You’ll understand more later…
   4. Drop the user on the Landing Page tab.
   5. Hide every sheet except for the Landing Page in case there is an error. This also gives a blank view to the user as the book loads in and manages information. Aesthetics…
   6. LudicrousMode is an optimization to help the book run faster in the next steps.
   7. The book then looks to see if a user has already opened the book before. When someone gets their book and opens it, their credentials are branded into the top left A column of the Data Tab. This is important for user tracking and is a key piece of how the book runs.
      1. If a user has already opened the book and branded it with their credentials, their M0 number will be in cell A3 on the data tab. To keep from overwriting this but to also give this user the correct experience, another module called seconduser contains a sub called seconduser. It will copy the current user (2nd, 3rd, 4th, nth) and put their credentials in the A column down on row 55.
      2. The book then runs the FindMatchingM0 sub in the TierUsers module. Ignore this for now. It’s more of a last step.
      3. If cell A3, which holds the initial openers M0, is empty, then run the book for the first time.
   8. There’s another if statement to see that A3 is empty again and then starts pulling in information.
   9. The employee information is pulled from the EMP\_MGR\_AttTest database and added to the data sheet by calling the queryEMP\_M0 sub. This is a SQL query that grabs all of the updated HR information so that user M0s can also be associated with their names, email addresses, managers, and some other information.
   10. Now that all employee information is available, the book will create a userID in the sub UserID by clearing anything that could be there, grabbing the employeeM0 which is the Environ(“USERNAME”) value. It will see if the user is a DEV and then, if they are, allow them to change the M0 number to test the book as someone else if they wish. Once the M0 is established, the user’s name is added as well as a template ID.

**We now know who the user is, but must construct the information for the user to play their part.**

* 1. The TreeBuild takes the recursion information and builds a hierarchical tree on the Data Tab using the M0 numbers.
  2. Now that the hierarchy is built, we need to grab the first user for each branch. This is whoever is the furthest to the right. These are the people who will make up the first tier of the attestation and are also all of the distinct delegates identified in the RCL. In the TierSystem module is a sub called Tierz that will build the tiers of users. It grabs first tier people by grabbing everyone on the right. Then it adds their boss for the next tier. The boss of the boss for tier 3, and so on and so forth until the top is reached. Blanks are removed to roll the tiers up and create complete columns. Managers need to only belong to the highest tier they are in, so a remove duplicates function is run to remove any records of that person in the lower tiers. This repeats from the highest tier down to the lowest (Right to Left). Finally, because the CEO doesn’t have a boss, the ‘none’ value is also removed to keep from complicating things.

**Everyone is organized into their tiers and we know who the bosses are. Next we need the questions.**

* 1. The questions for the process are stored in the ERMDB database in the AttQuest table. These values are pulled into the Data Tab to build the Questionnaire Tab.
  2. Anyone who has submitted their responses will have them stored in the ERMDB database in the AttResponse table. These responses are pulled into the Data Tab for reference throughout the process.
  3. A manager will have to review the responses of the people below them, so the sub ReportResponseBuild in the ReportResponse module counts the reports for the userID M0 number. It grabs all reports to manager, checks if they are in the tiers, and removes any that aren’t in a tier. This is because if they aren’t in a tier, they won’t be part of the process. It counts the reports the manager has and it counts how many reports have responded. If they match, then the manager will have permission to answer their questions because they can review their reports answers as they do.
  4. The QuestBuild sub builds the questions on the Questionnaire Tab.
  5. ReportReponseInsert takes the report answers from ReportResponseBuild and inserts them under each question on the Questionnaire tab. That way managers can review the answers and include pertinent information as they certify.
  6. A manager or SPOC may want to send an individual reminder email to someone. This takes the people and emails from the tier system and builds a total list of users so the dropdown for the reminder email will have the right names in it.
  7. ValidationQuestionnaire adds dropdown selections for question answers. These are usually user names for sending emails to those users.
  8. All of the pieces for a user to finish their attestation have been built, so now it’s time to see what page that person should be dropped onto. So far, all of the tabs have been hidden from the user during the loading of the previous steps. The TierUsers.FindingMatchingM0 identifies the user’s roll and unlocks and activates the tabs they should be using based on their credentials. It also checks for managers to see that all reports have responded before unlocking questions. This prevents managers from certifying before their reports.
  9. CompletionChecks.UserCompletionCheck looks to see if a user has already submitted answers. If they have, it locks them out of the questions sheet.

**And that is how the book opens…**

**Buttons and Functions**

At this point, the book is built and the user is dropped onto their respective pages. Below we will go through each page and what the buttons do.

1. The Data Tab has a RESET and RUN button. The reset button resets the book to a blank state, and the RUN button initiates the book opening. These are great for Developer testing.
2. The Progress Report tab is going to be seen by Owner / Delegates and Managers. Whoever falls into a tier gets this page.
   1. The user has the Individual Reminder button that uses the dropdown box next to it to select the recipient of the reminder email.
   2. The Group Reminder button goes through all report responses and sends a group reminder to anyone who hasn’t already responded to their certifications.
   3. The SPOC Hotline allows the user to directly email the SPOC group email to request assistance or a user reset.
3. The Questionnaire tab is where users will respond to questions using the dropdown selections and by filling in comments. Once all answers are filled in, the button at the top will record the answers and send a copy of the book to a SPOC for review and submission to the database.
4. The Admin Prog tab is for Finance and SPOCs to see where the process currently stands.
5. The Admin tab has a lot of buttons…
   1. Email Selected Tier sends the attestation initiation email to all users of the tier selected in the dropdown box to the right of it.
   2. Email Individual sends a reminder email to the user selected in the dropdown to the right of it.
   3. Reset Individual resets the user and all of their manager’s responses by selecting a user from the dropdown to the right of it.
   4. Push to DB is used when a book is received from a user and their answers need to be submitted to the AttResponse database. It checks to make sure a user doesn’t already have answers and blocks repeat submissions.
   5. Review Responses pulls all complete responses onto the Admin tab for review.
   6. Clear clears the Admin tab.
   7. Final Pull grabs all responses complete or not. This is for final review when all responses are in.
   8. Final Push takes the complete list of all responses and pushes them to the AttStorage database.
6. The Finance tab has 4 buttons that finance can use as they comment on user comments.
   1. Pull Comments pulls all submitted responses that have comments by the user.
   2. Save Changes saves the finance comments to the database.
   3. Clear clears the finance tab.
   4. SPOC Hotline allows finance to email the SPOCs to request assistance or a reset.

**Progress Reports**

The Progress Report and Admin Prog tabs run on activation to create the status colored user trees. The only difference between them is the Admin Prog tree is the entire hierarchy for attestation, whereas the Progress Report is only the reports who answer to the current workbook user. A user is red if they do have a completed Certification submission in the database. They are orange if they have submitted a certification, but they made comments which must be reviewed by finance. Once finance reviews their comments and leaves the finance comments on the record, then their name will turn green. If their certification did not have any free form comments, then

1. Admin Prog – Used by SPOCs and finance to understand the current status of all Owners/Delegates and their managers. It can be viewed by anyone who has SPOC, Finance, or DEV permissions.
   1. To build this magnificent piece of computer engineering marvel, the LudicrousMode is activated to make the calculations run faster without the user seeing everything being built.
   2. The AttResponses are pulled again so that the most up to date information is available for the tree coloring. If a SPOC or Finance person adds their comments or uploads, you want the tree to reflect those updates.
   3. The AdminTree.TreeBuild2 builds the hierarchical tree on the Admin Prog tab made of employee M0 numbers. It is referencing the recursion on the data tab, but building the tree on the Admin Prog tab.
   4. To get the colors correct, the comments need to be manipulated so that they can be referenced. The ProgRpt.MoveComs goes through the responses in the database and, if they have a comment, moves the user information to the comments column (IO) and also to the column that holds people needing finance comments (JD).
   5. ProgRpt.MoveFinComs loops through the comment names and checks the AttResponse values to see if there are finance comments for each comment made by the user. If there are finance comments for each user comment, the user name will be deleted from the finance comments column (JD).
   6. Replace the M0 numbers in the tree with the people’s actual name. The tree value that it loops through is too big… You could definitely optimize this. It then goes through the tree and colors the individual names based on whether or not that person is in the comments or finance columns IO or JD…. This is running off name right now…. Should probably be switched to M0…
   7. Turn of LudicrousMode.
2. Progress Report –
   1. The only difference between this one (User view of Report) and the Super User view above, is that in the ProgRpt.UserTreeResponseCheck the user is added and all reports that have them as a boss are added to the sheet. After adding all reports, the book checks if those reports are in a tier. If they aren’t they are dropped, because they don’t belong here. The colors are then applied to all responsible parties.