**SQRCT - Strategic Quote Recovery & Conversion Tracker**

**1. Project Overview**

**SQRCT** is a system designed to track and manage the follow-up process for sales quotes that have been issued but not yet converted or closed. Its primary goal is to improve quote conversion rates by providing visibility into outstanding quotes, automating follow-up stage calculation, and facilitating consistent engagement by the sales team.

This system operates within the **Microsoft Excel environment**, utilizing **Power Query (M)** for data ingestion/transformation and **Visual Basic for Applications (VBA)** for automation, user interface elements, and data synchronization logic.

**System Structure:**

The operational system consists of multiple interconnected Excel workbooks:

1. **User Workbooks:** Individual files (e.g., ...Ryan Working Version...xlsm, ...Ally Working Version...xlsx) where users interact with their version of the SQRCT Dashboard and record engagement details.
2. **Master Workbook:** A central file (...Automated Master Version...xlsx) intended to hold the consolidated, authoritative dataset after synchronization.
3. **SyncTool Workbook:** A separate Excel file (...TOOL\_SQRCT\_SyncTool...xlsm) containing VBA code manually triggered to read data from the User and Master workbooks, resolve conflicts, and write the merged results back to the Master workbook.

**Repository Purpose:**

This GitHub repository serves as the central location for:

* **Version Control:** Managing the source code (VBA modules, class modules, worksheet code, Power Query M scripts) extracted from the operational Excel files.
* **Documentation:** Hosting technical (ARCHITECTURE.md), user, and contribution guides (CONTRIBUTING.md).
* **Issue Tracking:** Reporting bugs and suggesting enhancements related to the VBA code, Power Query logic, or overall process.
* **Collaboration:** Providing a reference point for developers working on maintaining or improving the SQRCT system.

**Core Functionality:**

* Ingests quote data from daily CSV exports and historical records using Power Query.
* Calculates quote aging and determines automated follow-up stages (AutoStage) and notes (AutoNote) using Power Query.
* Provides a central view ("SQRCT Dashboard" sheet) populated via Power Query and VBA.
* Allows users (in their workbooks) to record manual engagement details (Engagement Phase, Last Contact Date, Email Contact, Comments) captured via VBA (Worksheet\_Change) into a hidden "UserEdits" sheet.
* Includes a VBA-based SyncTool (separate workbook) to merge "UserEdits" data from User/Master workbooks, resolve conflicts based on timestamps, and update the Master workbook's "UserEdits" sheet.

**2. Features**

* **Automated Data Ingestion & Processing:** Leverages Power Query to combine and transform data from CSVs and local tables.
* **Calculated Quote Status:** Determines AutoStage and AutoNote based on defined logic in Power Query.
* **User Edit Management:** Captures and stores user-provided updates locally within each user's workbook via VBA.
* **Conflict Resolution:** The SyncTool merges edits from multiple users, prioritizing the most recent updates (with special handling for comments).
* **Centralized Master Data:** The SyncTool consolidates quote status and user edits into the Master workbook.
* **Logging:** The SyncTool records synchronization activities and errors for traceability within its own workbook.

**3. Core Technologies**

* Microsoft Excel (Desktop Application)
* Power Query (M Language)
* Visual Basic for Applications (VBA)

**4. Prerequisites**

* Microsoft Excel (version supporting Power Query and VBA Macros)
* Access to the required network folders and file locations used by Power Query and the SyncTool (see Configuration).
* Git (for interacting with this repository).

**5. Working with the Repository Code**

This repository stores the *reference source code* extracted from the operational Excel files.

1. **Clone the repository:**
2. git clone <repository-url>
3. cd sqrct-repository # Or your chosen directory name
4. **Accessing Code:**
   * VBA code (.bas, .cls, .frm files) can be found in the relevant subdirectories (TBD - need to define where extracted code lives, e.g., /vba/user\_workbook, /vba/sync\_tool).
   * Power Query M code can be found (TBD - e.g., in /pq/ as text files or within exported workbook structures if using specific tooling).
5. **Applying Changes (Manual Process):**
   * Modifications made to the code in this repository need to be **manually imported/updated** into the corresponding operational .xlsm/.xlsx files.
   * Similarly, changes made directly in the Excel files should be extracted and committed back to this repository to maintain synchronization.
   * *(Note: Tooling exists to help automate VBA source control, but requires specific setup and is outside the scope of this basic README).*

**6. Configuration**

While the core logic resides in Excel, certain configurations are critical:

* **File Paths:** Power Query queries (CLIENT QUOTES, CSVQuotes) and the SyncTool VBA (Module\_File\_Processor, Module\_SyncTool\_UI) rely on specific file paths (e.g., S:\..., R:\..., paths selected in the SyncTool UI).
  + **Recommendation:** Avoid hardcoding paths directly in M code or VBA where possible. Consider using named ranges in a configuration sheet within the relevant workbooks or parameterizing Power Query steps if feasible. Paths configured via the SyncTool UI should be clearly documented.
* **Secrets:** Ensure no sensitive information (passwords, API keys) is stored directly in VBA code or M queries. The hardcoded "password" in the user workbook VBA should be addressed.

**7. Usage Workflow**

The standard operational workflow involves manual steps:

1. **User Work:** Ryan and Ally open their respective "Working Version" Excel files and make edits to columns K-N on the "SQRCT Dashboard". These edits are automatically saved to their local hidden "UserEdits" sheet by VBA. They may also manually refresh Power Query data as needed.
2. **Synchronization:** A designated user opens the "SyncTool" Excel workbook.
   * They use the "Browse" buttons on the SyncTool Dashboard to select the current file paths for Ryan's, Ally's, and the Master workbooks.
   * They click the "Sync" button.
   * The SyncTool VBA code executes: reads data from all three "UserEdits" sheets, resolves conflicts, and writes the merged data back to the Master workbook's "UserEdits" sheet.
   * The SyncTool then triggers a refresh in the Master workbook.
3. **Master Update:** The refresh in the Master workbook updates its Power Query queries and runs its RefreshDashboard VBA, populating its dashboard with the latest core data and the newly merged user edits.

**8. Testing**

* Currently, testing relies heavily on **manual procedures**:
  + Verifying Power Query output for different source data scenarios.
  + Testing VBA functionality by clicking buttons and entering data in user workbooks and the SyncTool.
  + Manually comparing outputs after synchronization.
* The /tests directory in this repository can be used to store manual test plans, checklists, or sample data used for testing.
* *(Note: Automated testing for VBA/Power Query within Excel is challenging and typically requires external tools or frameworks not currently implemented.)*

**9. Contribution**

We welcome contributions to improve the SQRCT system! Please read our CONTRIBUTING.md file for details on:

* Development environment considerations (Excel versions, VBA editor settings).
* VBA and Power Query M coding standards.
* Branching strategy for managing code changes in Git.
* Commit message format.
* Pull request process for reviewing code changes intended for the repository.
* Issue reporting using the GitHub Issues tab.

**10. License**

This project is licensed under the MIT License. See the LICENSE file for full details.

**11. Support & Contact**

* **Issues:** If you encounter a bug or have a feature request related to the code or process, please submit an issue through the GitHub repository's "Issues" tab, using the provided templates.
* **Contact:** For specific questions or support, please contact [Your Name/Team Contact] at [your-email@example.com] or relevant internal channel.