--- FILE: ImplementationPlan.md ---

Implementation Plan - The Strategy Board Upgrade

Version: 1.0.0 **Target:** 0.105.0-Beta

1. Context

This plan executes the "Strategy Board" UI upgrade defined in the Phase 4 pivot. We are surfacing "Run QSOs" and "Run %" metrics to the main dashboard table to provide immediate strategic insight.

2. Proposed Changes

A. `contest_tools/data_aggregators/time_series.py`

Goal: Calculate scalar Run/S&P metrics in the aggregator so they are available for the Dashboard without re-processing.

- **Logic:**
  - Locate the scalar calculation block (where `net_qsos` is defined).
  - Filter the valid (non-dupe) DataFrame for `Run == 'Run'`.
  - Calculate `scalar_run_qsos` and `scalar_run_percent`.
  - Add these keys to the `log_entry["scalars"]` dictionary.

B. `web_app/analyzer/views.py`

Goal: Extract the new scalars and pass them to the template context.

- **Logic:**
  - Update the `context['logs']` construction loop.
  - Map `data['scalars']['run_qsos']` and `data['scalars']['run_percent']` to the template dictionary.

C. `web_app/analyzer/templates/analyzer/dashboard.html`

Goal: Display the new metrics in the "Arena" table.

- **Logic:**
  - Add `<th>Run QSOs</th>` and `<th>Run %</th>` headers.
  - Add corresponding `<td>` cells in the loop.
  - Apply Bootstrap styling (`text-end`) for alignment.

3. Surgical Change Verification

File: `contest_tools/data_aggregators/time_series.py`

[cite_start]**Baseline:** Version 1.5.0 [cite: 1552] **Changes:** ___CODE_BLOCK___diff
--- contest_tools/data_aggregators/time_series.py +++ contest_tools/data_aggregators/time_series.py
@@ -15,4 +15,7 @@

--- Revision History ---

+# [1.6.0] - 2025-12-13 +# - Added `run_qsos` and `run_percent` to scalar output for Strategy Board dashboard.

[1.5.0] - 2025-12-12

- Added final_score and mult_breakdown to the scalar output to support

high-fidelity dashboards.

@@ -58,4 +61,6 @@ points_sum = 0 year = ”UnknownYear”

- `            scalar_run_qsos = 0`
- `            scalar_run_percent = 0.0`
```
        else:
            gross_qsos = len(df_full)
```

@@ -66,4 +71,11 @@

- `            # Calculate Scalar Run Metrics (Net QSOs only)`
- `            df_valid_scalars = df_full[df_full['Dupe'] == False]`
- `            scalar_run_qsos = df_valid_scalars[df_valid_scalars['Run'] == 'Run'].shape[C`
- `            scalar_run_percent = (scalar_run_qsos / net_qsos * 100) if net_qsos > 0 else`
- `            # Extract Year`
```
            date_series = df_full['Date'].dropna()
```

@@ -93,5 +105,7 @@ ”contest_name”: str(contest_name), ”year”: str(year), ”final_score”: int(final_score_scalar),

- `                "mult_breakdown": mult_breakdown`
- `                "mult_breakdown": mult_breakdown,`
- `                "run_qsos": int(scalar_run_qsos),`

-
  ```
                "run_percent": round(float(scalar_run_percent), 1)
          },
          "cumulative": {
  ```

CODE_BLOCK

File: `web_app/analyzer/views.py`

[cite_start]**Baseline:** Version 0.104.0-Beta [cite: 1878] **Changes:** ___CODE_BLOCK___diff --- web_app/analyzer/views.py +++ web_app/analyzer/views.py @@ -15,4 +15,7 @@

--- Revision History ---

+# [0.105.0-Beta] - 2025-12-13 +# - Added 'run_qsos' and 'run_percent' to the dashboard context context.

[0.104.0-Beta] - 2025-12-12

- Updated analyze_logs to consume new 'final_score' and 'mult_breakdown'

from TimeSeriesAggregator instead of raw points.

@@ -113,5 +116,7 @@ 'callsign': call, 'score': data['scalars'].get('final_score', 0), 'qsos': data['scalars']['net_qsos'],

- `                    'mults': data['scalars'].get('mult_breakdown', {})`
- `                    'mults': data['scalars'].get('mult_breakdown', {}),`
- `                    'run_qsos': data['scalars'].get('run_qsos', 0),`

-
  ```
                      'run_percent': data['scalars'].get('run_percent', 0.0)
                  })
  ```

CODE_BLOCK

File: `web_app/analyzer/templates/analyzer/dashboard.html`

[cite_start]**Baseline:** from project bundle [cite: 1898] **Changes:** ___CODE_BLOCK___diff --- web_app/analyzer/templates/analyzer/dashboard.html +++ web_app/analyzer/templates/analyzer/dashl
@@ -14,4 +14,6 @@ Callsign Total QSOs

- `                    <th class="py-3 text-end">Run QSOs</th>`

-
  ```
  <th class="py-3 text-end">Run %</th>
  {% for header in mult_headers %}
  <th class="py-3 text-end">{{ header }}</th>
  ```

@@ -23,4 +25,6 @@ {{ log.callsign }} {{ log.qsos }}

-
  ```
  <td class="text-end fw-bold">{{ log.run_qsos }}</td>
  ```

-
  ```
  <td class="text-end">{{ log.run_percent }}%</td>
  {% for header in mult_headers %}
  <td class="text-end">{{ log.mults|get_item:header }}</td>
  ```

CODE_BLOCK

4. Pre-Flight Check

- **Consistency:** The aggregator calculates the %; the view passes it; the template renders it. Logic flow is sound.
- **Validation:**
  - `time_series.py`: Using `df_valid_scalars` ensures we don't count Dupes in the Run %, matching the Net QSOs denominator.
  - `views.py`: `.get()` method defaults prevent errors if the aggregator version is mismatched (though we are updating both).
- **Visuals:** Columns added in the correct order (between Total QSOs and Mults).

5. Next Steps

1. User creates `builder_bundle.txt`.
2. User triggers ”Act as Builder”.