Power BI Implementation Guide
Hot Rolling Plant - Temper Line Analytics
ArcelorMittal Vanderbijlpark Works

Document Overview

This guide provides complete Power BI implementation instructions for the Hot Rolling Plant analytics system developed to address the 30% production increase following the Saldanha Works closure. The system integrates production, maintenance, and equipment data to deliver actionable insights on tempo, bottlenecks, downtime, and shift performance.

Project Context:
- 30% increase in monthly production targets after Saldanha closure
- Temper line not originally designed for thin flat products
- No historical data available for new product mix
- Falling tempo, shift delays, and frequent breakdowns
- Need for data-driven operational intelligence

Business Impact Achieved:
- ~10% tempo improvement through bottleneck identification
- ~5% reduction in maintenance downtime
- Reduced shift losses (led to 4-shift to 3-shift recommendation)
- Consistent achievement of monthly production targets
- 40% reduction in manual reporting effort

---

1. Data Model Setup

1.1 Import Data Tables

Import all 8 CSV files from the data pipeline into Power BI:

Dimension Tables:
1. `dim_equipment.csv`
2. `dim_date_crew_schedule.csv`

Fact Tables:
3. `fact_production_coil.csv` (Real MES completion times)
4. `fact_maintenance_event.csv`
5. `fact_coil_operation_cycle.csv` (Equipment-level operations)
6. `fact_equipment_event_log.csv` (RUN/IDLE/FAULT timeline)

Reference Tables (Optional):

7. `raw_production_filtered.csv`
8. `raw_maintenance_filtered.csv`

1.2 Data Model Relationships

Create the following relationships in Power BI Model view:

Primary Relationships:

1. **fact_production_coil → dim_date_crew_schedule**
   - From: `fact_production_coil[production_date]`
   - To: `dim_date_crew_schedule[production_date]`
   - Cardinality: Many-to-One
   - Cross filter: Single

2. **fact_coil_operation_cycle → dim_equipment**
   - From: `fact_coil_operation_cycle[equipment_id]`
   - To: `dim_equipment[equipment_id]`
   - Cardinality: Many-to-One
   - Cross filter: Both

3. **fact_coil_operation_cycle → fact_production_coil**
   - From: `fact_coil_operation_cycle[coil_id]`
   - To: `fact_production_coil[coil_id]`
   - Cardinality: Many-to-One
   - Cross filter: Both

4. **fact_equipment_event_log → dim_equipment**
   - From: `fact_equipment_event_log[equipment_id]`
   - To: `dim_equipment[equipment_id]`
   - Cardinality: Many-to-One
   - Cross filter: Both

5. **fact_maintenance_event → dim_equipment**
   - From: `fact_maintenance_event[equipment_name]`
   - To: `dim_equipment[equipment_name]`
   - Cardinality: Many-to-One
   - Cross filter: Single

Note: Set up inactive relationships where multiple paths exist. Activate them in specific measures using USERELATIONSHIP().

1.3 Data Type Verification

Ensure correct data types for all columns:

Dates and Times:

- `production_date`, `event_date` → Date
- `completion_ts`, `start_datetime`, `end_datetime` → DateTime
- `operation_start_ts`, `operation_end_ts` → DateTime

Numeric Fields:
- `thickness_mm`, `width_mm`, `mass_out_tons` → Decimal Number
- `total_cycle_time_min`, `duration_hours`, `duration_min` → Decimal Number
- `operation_duration_sec`, `event_duration_sec` → Decimal Number
- `gap_from_prev_completion_min`, `gap_from_prev_parent_min` → Decimal Number

Text Fields:
- `coil_id`, `parent_coil_id`, `equipment_name`, `type_code`, `shift_code` → Text

Boolean Fields:
- `is_prime`, `is_scrap`, `is_bottleneck_candidate`, `is_active` → True/False

---

2. Core Measure Library

Create a new table called "Measures" to organize all DAX measures. Use CTRL+K to format DAX code.

2.1 Production Metrics

```dax
// Total Pieces Produced
Total Pieces = COUNTROWS(fact_production_coil)

// Prime Pieces
Prime Pieces =
CALCULATE(
    COUNTROWS(fact_production_coil),
    fact_production_coil[is_prime] = TRUE
)

// Scrap Pieces
Scrap Pieces =
CALCULATE(
    COUNTROWS(fact_production_coil),
    fact_production_coil[is_scrap] = TRUE
)

// Prime Rate %
Prime Rate % =
DIVIDE(
    [Prime Pieces],
    [Total Pieces],
```

```
    0
) * 100

// Total Parent Coils
Total Parent Coils = DISTINCTCOUNT(fact_production_coil[parent_coil_id])

// Pieces per Parent Coil
Pieces per Parent =
DIVIDE(
    [Total Pieces],
    [Total Parent Coils],
    0
)

// Average Cycle Time (minutes)
Avg Cycle Time = AVERAGE(fact_production_coil[total_cycle_time_min])

// Total Mass Output (tons)
Total Mass Output = SUM(fact_production_coil[mass_out_tons])

// Pieces per Hour (Throughput)
Pieces per Hour =
DIVIDE(
    60,
    [Avg Cycle Time],
    0
)
```

2.2 Tempo and Gap Metrics

```dax
// Average Completion Gap (minutes between pieces)
Avg Completion Gap =
AVERAGE(fact_production_coil[gap_from_prev_completion_min])

// Average Parent Gap (minutes between parent coils)
Avg Parent Gap =
AVERAGE(fact_production_coil[gap_from_prev_parent_min])

// Median Completion Gap
Median Completion Gap =
MEDIAN(fact_production_coil[gap_from_prev_completion_min])

// Tempo Target (pieces/hour) - Set based on plant target
Tempo Target = 4.0  // Adjust based on your target
```

```dax
// Tempo Achievement %
Tempo Achievement % =
DIVIDE(
    [Pieces per Hour],
    [Tempo Target],
    0
) * 100

// Tempo Variance (pieces/hour)
Tempo Variance = [Pieces per Hour] - [Tempo Target]
```

2.3 Equipment Utilization Metrics

```dax
// Total RUN Time (hours)
Total RUN Hours =
CALCULATE(
    SUM(fact_equipment_event_log[event_duration_sec]) / 3600,
    fact_equipment_event_log[event_type] = "RUN"
)

// Total IDLE Time (hours)
Total IDLE Hours =
CALCULATE(
    SUM(fact_equipment_event_log[event_duration_sec]) / 3600,
    fact_equipment_event_log[event_type] = "IDLE"
)

// Total FAULT Time (hours)
Total FAULT Hours =
CALCULATE(
    SUM(fact_equipment_event_log[event_duration_sec]) / 3600,
    fact_equipment_event_log[event_type] = "FAULT"
)

// Total Equipment Time (hours)
Total Equipment Hours =
[Total RUN Hours] + [Total IDLE Hours] + [Total FAULT Hours]

// Equipment Utilization %
Equipment Utilization % =
DIVIDE(
    [Total RUN Hours],
    [Total Equipment Hours],
    0
) * 100
```

```
// Equipment Availability % (RUN + IDLE)
Equipment Availability % =
DIVIDE(
    [Total RUN Hours] + [Total IDLE Hours],
    [Total Equipment Hours],
    0
) * 100

// Downtime % (FAULT only)
Downtime % =
DIVIDE(
    [Total FAULT Hours],
    [Total Equipment Hours],
    0
) * 100
```

2.4 Bottleneck Analysis Metrics

```dax
// Bottleneck Equipment Operation Time (hours)
Bottleneck Operation Hours =
CALCULATE(
    SUM(fact_coil_operation_cycle[operation_duration_sec]) / 3600,
    dim_equipment[is_bottleneck_candidate] = TRUE
)

// Total Operation Time (hours)
Total Operation Hours =
SUM(fact_coil_operation_cycle[operation_duration_sec]) / 3600

// Bottleneck Share %
Bottleneck Share % =
DIVIDE(
    [Bottleneck Operation Hours],
    [Total Operation Hours],
    0
) * 100

// Average Equipment Operation Time (minutes)
Avg Equipment Operation Time =
AVERAGE(fact_coil_operation_cycle[operation_duration_sec]) / 60

// Equipment Bottleneck Severity Score
Equipment Bottleneck Score =
VAR EquipmentOpsTime =
```

```
    SUM(fact_coil_operation_cycle[operation_duration_sec])
VAR TotalOpsTime =
    CALCULATE(
        SUM(fact_coil_operation_cycle[operation_duration_sec]),
        ALL(dim_equipment)
    )
VAR TimeShare = DIVIDE(EquipmentOpsTime, TotalOpsTime, 0)
VAR IsBottleneck = MAX(dim_equipment[is_bottleneck_candidate])
RETURN
    IF(IsBottleneck, TimeShare * 100, TimeShare * 50)
```

2.5 Maintenance Metrics

```dax
// Total Maintenance Events
Total Maintenance Events = COUNTROWS(fact_maintenance_event)

// Total Maintenance Downtime (hours)
Total Maintenance Hours = SUM(fact_maintenance_event[duration_hours])

// Average Downtime per Event (hours)
Avg Downtime per Event =
DIVIDE(
    [Total Maintenance Hours],
    [Total Maintenance Events],
    0
)

// Unplanned Downtime (hours) - Adjust categories as needed
Unplanned Downtime Hours =
CALCULATE(
    SUM(fact_maintenance_event[duration_hours]),
    fact_maintenance_event[Category] <> "Planned downtime"
)

// Planned Downtime (hours)
Planned Downtime Hours =
CALCULATE(
    SUM(fact_maintenance_event[duration_hours]),
    fact_maintenance_event[Category] = "Planned downtime"
)

// MTBF - Mean Time Between Failures (hours)
MTBF =
VAR TotalProductionHours = [Total RUN Hours] + [Total IDLE Hours]
VAR FailureCount =
```

```
    CALCULATE(
        COUNTROWS(fact_maintenance_event),
        fact_maintenance_event[Category] <> "Planned downtime"
    )
RETURN
    DIVIDE(TotalProductionHours, FailureCount, 0)

// MTTR - Mean Time To Repair (hours)
MTTR =
CALCULATE(
    AVERAGE(fact_maintenance_event[duration_hours]),
    fact_maintenance_event[Category] <> "Planned downtime"
)
```

2.6 Shift Performance Metrics

```dax
// Pieces by Shift
Pieces This Shift =
CALCULATE(
    [Total Pieces],
    USERELATIONSHIP(
        fact_production_coil[production_date],
        dim_date_crew_schedule[production_date]
    )
)

// Shift Tempo (pieces/hour)
Shift Tempo =
VAR ShiftPieces = [Pieces This Shift]
VAR ShiftHours = 12  // Assuming 12-hour shifts
RETURN
    DIVIDE(ShiftPieces, ShiftHours, 0)

// Shift Efficiency vs Target %
Shift Efficiency % =
DIVIDE(
    [Shift Tempo],
    [Tempo Target],
    0
) * 100

// Best Performing Shift
Best Shift =
TOPN(
    1,
```

```
    SUMMARIZE(
        fact_production_coil,
        fact_production_coil[shift_code],
        "ShiftTempo", [Shift Tempo]
    ),
    [ShiftTempo],
    DESC
)

// Shift Handover Loss Estimate (minutes)
Shift Handover Loss =
VAR AvgGapOverall = [Avg Completion Gap]
VAR ShiftStartGaps =
    CALCULATE(
        AVERAGE(fact_production_coil[gap_from_prev_completion_min]),
        fact_production_coil[completion_ts].[Hour] >= 6,
        fact_production_coil[completion_ts].[Hour] < 7
    )
RETURN
    ShiftStartGaps - AvgGapOverall
```

2.7 Product Mix Metrics

```dax
// Thin & Narrow Pieces (Fast Band)
Fast Band Pieces =
CALCULATE(
    COUNTROWS(fact_production_coil),
    fact_production_coil[thickness_mm] <= 2.0,
    fact_production_coil[width_mm] <= 1300
)

// Other Mix Pieces
Other Mix Pieces =
[Total Pieces] - [Fast Band Pieces]

// Fast Band %
Fast Band % =
DIVIDE(
    [Fast Band Pieces],
    [Total Pieces],
    0
) * 100

// Average Cycle Time - Fast Band
Avg Cycle Fast Band =
```

```
CALCULATE(
    AVERAGE(fact_production_coil[total_cycle_time_min]),
    fact_production_coil[thickness_mm] <= 2.0,
    fact_production_coil[width_mm] <= 1300
)

// Average Cycle Time - Other Mix
Avg Cycle Other Mix =
CALCULATE(
    AVERAGE(fact_production_coil[total_cycle_time_min]),
    NOT(
        fact_production_coil[thickness_mm] <= 2.0 &&
        fact_production_coil[width_mm] <= 1300
    )
)

// Product Mix Complexity Score
Product Mix Complexity =
VAR UniqueWidths = DISTINCTCOUNT(fact_production_coil[width_mm])
VAR UniqueThickness = DISTINCTCOUNT(fact_production_coil[thickness_mm])
VAR UniqueGrades = DISTINCTCOUNT(fact_production_coil[Grade])
RETURN
    (UniqueWidths * 0.4) + (UniqueThickness * 0.4) + (UniqueGrades * 0.2)
```

2.8 Time Intelligence Measures

```dax
// Previous Period Total Pieces
Previous Period Pieces =
CALCULATE(
    [Total Pieces],
    DATEADD(fact_production_coil[production_date], -1, MONTH)
)

// Pieces MoM Change
Pieces MoM Change =
[Total Pieces] - [Previous Period Pieces]

// Pieces MoM % Change
Pieces MoM % =
DIVIDE(
    [Pieces MoM Change],
    [Previous Period Pieces],
    0
) * 100
```

```
// Rolling 7-Day Average Tempo
Rolling 7D Tempo =
CALCULATE(
    [Pieces per Hour],
    DATESINPERIOD(
        fact_production_coil[production_date],
        LASTDATE(fact_production_coil[production_date]),
        -7,
        DAY
    )
)

// Month-to-Date Pieces
MTD Pieces =
CALCULATE(
    [Total Pieces],
    DATESMTD(fact_production_coil[production_date])
)

// Year-to-Date Pieces
YTD Pieces =
CALCULATE(
    [Total Pieces],
    DATESYTD(fact_production_coil[production_date])
)
```

---

3. Calculated Columns

Add these calculated columns to enhance analysis capabilities:

3.1 In fact_production_coil

```dax
// Product Band Classification
Product Band =
IF(
    fact_production_coil[thickness_mm] <= 2.0 &&
    fact_production_coil[width_mm] <= 1300,
    "Fast Band (Thin & Narrow)",
    "Standard Mix"
)

// Product Type Category
Product Category =
```

```
SWITCH(
    TRUE(),
    fact_production_coil[is_prime] = TRUE, "Prime",
    fact_production_coil[is_scrap] = TRUE, "Scrap",
    "Other"
)

// Shift Period
Shift Period =
VAR CompletionHour = HOUR(fact_production_coil[completion_ts])
RETURN
    IF(CompletionHour >= 6 && CompletionHour < 18, "Day", "Night")

// Cycle Time Category
Cycle Time Category =
SWITCH(
    TRUE(),
    fact_production_coil[total_cycle_time_min] < 10, "Fast (<10 min)",
    fact_production_coil[total_cycle_time_min] < 15, "Normal (10-15 min)",
    fact_production_coil[total_cycle_time_min] < 20, "Slow (15-20 min)",
    "Very Slow (>20 min)"
)

// Gap Category
Gap Category =
SWITCH(
    TRUE(),
    ISBLANK(fact_production_coil[gap_from_prev_completion_min]), "First Piece",
    fact_production_coil[gap_from_prev_completion_min] < 2, "Continuous (<2 min)",
    fact_production_coil[gap_from_prev_completion_min] < 5, "Normal (2-5 min)",
    fact_production_coil[gap_from_prev_completion_min] < 10, "Gap (5-10 min)",
    "Long Delay (>10 min)"
)
```

3.2 In dim_equipment

```dax
// Bottleneck Status
Bottleneck Status =
IF(
    dim_equipment[is_bottleneck_candidate] = TRUE,
    "Bottleneck",
    "Non-Bottleneck"
)

// Equipment Priority Score
```

```
Equipment Priority =
SWITCH(
    TRUE(),
    dim_equipment[is_bottleneck_candidate] = TRUE &&
        dim_equipment[section] = "CENTRE", 3,
    dim_equipment[is_bottleneck_candidate] = TRUE, 2,
    1
)
```

---

4. Dashboard Design Specifications

Create 5 primary dashboards as separate pages in Power BI.

Dashboard 1: Executive Summary (Home)

Purpose: High-level KPIs for management and executives
Refresh: Real-time or hourly
Audience: Plant Manager, Production Manager, Executives

Layout Components:

Row 1: Key Performance Indicators (Card Visuals)
- Total Pieces (Current Month)
  - Measure: [MTD Pieces]
  - Comparison: vs Previous Month
  - Conditional formatting: Green if > target, Red if < target

- Prime Rate %
  - Measure: [Prime Rate %]
  - Target: 85% (adjust based on plant standards)
  - Format: Percentage with 1 decimal

- Average Tempo (Pieces/Hour)
  - Measure: [Pieces per Hour]
  - Comparison: vs Target tempo
  - Show variance: [Tempo Variance]

- Equipment Utilization %
  - Measure: [Equipment Utilization %]
  - Target: 75%
  - Color coding: >80% Green, 60-80% Yellow, <60% Red

Row 2: Trend Analysis

Left (60% width):
- Line Chart: "Daily Production Tempo Trend"
  - X-axis: production_date
  - Y-axis: [Pieces per Hour]
  - Add constant line: [Tempo Target]
  - Trend line: ON
  - Data labels: Top values only
  - Drill-down: Enabled (Date → Week → Day)

Right (40% width):
- Donut Chart: "Product Mix Distribution"
  - Values: [Total Pieces]
  - Legend: Product Category (Prime/Scrap)
  - Data labels: Percentage
  - Colors: Green (Prime), Red (Scrap)

Row 3: Performance Breakdown

Left:
- Clustered Column Chart: "Shift Performance Comparison"
  - X-axis: shift_code (A, B, C, D)
  - Y-axis: [Shift Tempo]
  - Add: [Tempo Target] as constant line
  - Data labels: ON
  - Sort: Descending by tempo

Middle:
- Stacked Bar Chart: "Equipment Utilization Breakdown"
  - Y-axis: equipment_name (Top 10)
  - X-axis (stacked): [Total RUN Hours], [Total IDLE Hours], [Total FAULT Hours]
  - Colors: Green (RUN), Yellow (IDLE), Red (FAULT)
  - Sort: By [Total FAULT Hours] descending
  - Filter: is_active = TRUE

Right:
- KPI Visual: "Monthly Target Achievement"
  - Value: [MTD Pieces]
  - Goal: Monthly Target (set as parameter)
  - Trend: [Previous Period Pieces]
  - Format: Auto-scaling

Row 4: Critical Alerts

- Table: "Top Equipment Issues"
  - Columns:
    - equipment_name
    - [Total FAULT Hours]

    - [Total Maintenance Events]
    - [Downtime %]
  - Filter: [Downtime %] > 10
  - Top N: 5
  - Conditional formatting: Red gradient on Downtime %

Slicers (Left Panel):
- Date Range (production_date)
- Shift Code (shift_code)
- Equipment Section (dim_equipment[section])

Dashboard 2: Bottleneck Analysis

Purpose: Deep-dive into equipment bottlenecks and constraints
Audience: Process Engineers, Production Supervisors

Layout Components:

Row 1: KPI Cards
- Bottleneck Operation Hours
- Bottleneck Share %
- Total Operation Hours
- Average Equipment Cycle Time

Row 2: Bottleneck Identification

Main Visual (70% width):
- Waterfall Chart: "Equipment Time Contribution (Bottleneck Cascade)"
  - Category: equipment_name (sorted by process_order)
  - Y-axis: [Avg Equipment Operation Time]
  - Show connector lines
  - Highlight bottleneck equipment (red)
  - Total bar at end

Side Panel (30% width):
- Matrix: "Bottleneck Severity Matrix"
  - Rows: equipment_name
  - Values:
    - [Equipment Bottleneck Score]
    - [Total Operation Hours]
    - Bottleneck Status
  - Conditional formatting: Heat map on score
  - Sort: Score descending

Row 3: Time Analysis

Left:
- Stacked Area Chart: "Cumulative Operation Time by Equipment"
  - X-axis: production_date
  - Y-axis: [Total Operation Hours]
  - Legend: equipment_name (Top 5 bottlenecks only)
  - Tooltip: Add [Equipment Utilization %]

Right:
- Scatter Chart: "Equipment Operation vs Idle Time"
  - X-axis: [Total RUN Hours]
  - Y-axis: [Total IDLE Hours]
  - Size: [Total FAULT Hours]
  - Legend: equipment_name
  - Labels: Show for bottleneck equipment only
  - Quadrant lines: Add median lines

Row 4: Detailed Analysis

- Table: "Equipment Operation Detail"
  - Columns:
    - equipment_name
    - section
    - [Avg Equipment Operation Time]
    - [Equipment Utilization %]
    - [Bottleneck Share %]
    - [Total Operation Hours]
  - Filter: is_bottleneck_candidate = TRUE
  - Conditional formatting:
    - Red: [Bottleneck Share %] > 15
    - Yellow: 10-15%
    - Green: < 10%

Filters:
- Date Range
- Shift Code
- Equipment Section
- Bottleneck Status (Yes/No)
- Product Band

Dashboard 3: Maintenance & Downtime

Purpose: Track maintenance events and their impact on production
Audience: Maintenance Manager, Reliability Engineers

Layout Components:

Row 1: Downtime KPIs
- Total Maintenance Hours
- Unplanned Downtime Hours
- MTBF (Mean Time Between Failures)
- MTTR (Mean Time To Repair)
- Total Maintenance Events

Row 2: Downtime Trend

Main Chart (100% width):
- Combo Chart: "Daily Downtime vs Production"
  - X-axis: production_date
  - Column Y-axis: [Total Maintenance Hours]
  - Line Y-axis: [Total Pieces]
  - Show: 7-day moving average for both
  - Highlight: Days with >8 hours downtime

Row 3: Equipment-Level Analysis

Left (50%):
- Horizontal Bar Chart: "Downtime by Equipment (Top 10)"
  - Y-axis: equipment_name
  - X-axis: [Total Maintenance Hours]
  - Data labels: ON
  - Sort: Descending
  - Colors: Gradient from yellow to red

Right (50%):
- Matrix: "Maintenance Event Breakdown"
  - Rows: equipment_name
  - Columns: fact_maintenance_event[Category]
  - Values: [Total Maintenance Hours]
  - Subtotals: ON
  - Conditional formatting: Color scale

Row 4: Event Details

Left (60%):
- Scatter Chart: "MTBF vs MTTR Analysis"
  - X-axis: [MTBF]
  - Y-axis: [MTTR]
  - Size: [Total Maintenance Events]
  - Legend: equipment_name
  - Quadrants:
    - Top-right: "Frequent & Long" (Critical)

  - Top-left: "Rare but Long" (Monitor)
  - Bottom-right: "Frequent but Short" (Optimize)
  - Bottom-left: "Rare & Short" (Good)

Right (40%):
- Table: "Recent Critical Events"
  - Columns:
    - fact_maintenance_event[Start]
    - equipment_name
    - fact_maintenance_event[Category]
    - fact_maintenance_event[duration_hours]
  - Top N: 20
  - Sort: By Start date descending
  - Filter: duration_hours > 2

Row 5: Downtime Impact

- Ribbon Chart: "Maintenance Category Trends"
  - X-axis: production_date (Month)
  - Y-axis: [Total Maintenance Hours]
  - Ribbon: fact_maintenance_event[Category]
  - Shows: How maintenance patterns change over time

Filters:
- Date Range
- Equipment Name
- Maintenance Category
- Crew
- Minimum Duration (hours)

Dashboard 4: Shift Performance

Purpose: Compare shift performance and identify handover issues
Audience: Shift Supervisors, Production Coordinators

Layout Components:

Row 1: Shift Comparison KPIs

- Cards (One per shift: A, B, C, D):
  - Shift Tempo
  - Pieces This Shift
  - Shift Efficiency %
  - Average Cycle Time
  - Prime Rate %

Row 2: Shift Trends

Main Visual (100% width):
- Line Chart: "Shift Performance Over Time"
  - X-axis: production_date
  - Y-axis: [Shift Tempo]
  - Legend: shift_code (4 lines: A, B, C, D)
  - Markers: ON
  - Target line: [Tempo Target]
  - Smoothing: 3-day moving average

Row 3: Comparative Analysis

Left (40%):
- Clustered Bar Chart: "Shift Efficiency Comparison"
  - Y-axis: shift_code
  - X-axis: [Shift Efficiency %]
  - Data labels: Percentage
  - Reference line: 100% (target)
  - Colors: Conditional formatting

Middle (30%):
- Donut Chart: "Production Distribution by Shift"
  - Values: [Total Pieces]
  - Legend: shift_code
  - Data labels: Percentage + Value
  - Center label: [Total Pieces]

Right (30%):
- Card with sparkline: "Best Performing Shift"
  - Main value: shift_code with highest tempo
  - Sparkline: [Shift Tempo] trend
  - Subtitle: [Shift Efficiency %]

Row 4: Shift Handover Analysis

Left (50%):
- Line Chart: "Hourly Tempo Pattern (24-Hour View)"
  - X-axis: Hour of Day (0-23)
  - Y-axis: [Pieces per Hour]
  - Highlight: Shift change times (6:00, 18:00)
  - Show: Average tempo by hour
  - Annotations: Mark shift changes

Right (50%):
- Column Chart: "Shift Handover Gaps"

  - X-axis: Shift Transition (Day→Night, Night→Day)
  - Y-axis: [Shift Handover Loss]
  - Data labels: Minutes lost
  - Colors: Red if > 10 min, Yellow 5-10, Green < 5

Row 5: Crew Performance Detail

- Table: "Detailed Shift Metrics"
  - Rows: shift_code
  - Columns:
    - [Pieces This Shift]
    - [Shift Tempo]
    - [Avg Cycle Time]
    - [Prime Rate %]
    - [Shift Efficiency %]
  - Totals: Average for all shifts
  - Conditional formatting:
    - Icon set for efficiency
    - Color gradient for tempo

Filters:
- Date Range
- Shift Code (multi-select)
- Shift Period (Day/Night)
- Day of Week
- Product Band

Dashboard 5: Product Mix & Tempo

Purpose: Analyze how product characteristics affect cycle time and tempo
Audience: Process Engineers, Production Planners

Layout Components:

Row 1: Product Mix KPIs
- Total Pieces
- Fast Band Pieces
- Fast Band %
- Avg Cycle Fast Band
- Avg Cycle Other Mix
- Product Mix Complexity

Row 2: Product Mix Distribution

Left (50%):

- Scatter Chart: "Width vs Thickness (Cycle Time Bubble)"
  - X-axis: thickness_mm
  - Y-axis: width_mm
  - Size: total_cycle_time_min
  - Color: Product Band
  - Tooltip: Add coil_id, type_code
  - Zoom sliders: Enabled
  - Quadrants: Mark Fast Band region

Right (50%):
- Histogram: "Cycle Time Distribution"
  - X-axis: Cycle Time bins (5-min intervals)
  - Y-axis: Count of pieces
  - Color: Product Band
  - Overlaid curves: Normal distribution
  - Reference lines: Mean, Target

Row 3: Cycle Time Analysis

Left (60%):
- Box and Whisker Plot: "Cycle Time by Product Type"
  - Category: type_code (HL, HX, 98, etc.)
  - Values: total_cycle_time_min
  - Show: Outliers
  - Sort: By median
  - Color: Prime vs Scrap

Right (40%):
- Matrix: "Average Cycle Time Matrix"
  - Rows: Thickness ranges (<2mm, 2-3mm, >3mm)
  - Columns: Width ranges (<1300, 1300-1500, >1500)
  - Values: [Avg Cycle Time]
  - Conditional formatting: Green (fast) to Red (slow)
  - Show: Piece count in tooltip

Row 4: Tempo Impact

Full Width:
- Combo Chart: "Daily Product Mix vs Tempo"
  - X-axis: production_date
  - Column Y-axis (stacked): [Fast Band Pieces], [Other Mix Pieces]
  - Line Y-axis: [Pieces per Hour]
  - Show: Correlation between mix and tempo
  - Trend line: ON for tempo

Row 5: Parent Coil Analysis

Left (50%):
- Column Chart: "Pieces per Parent Coil Distribution"
  - X-axis: Number of pieces (1, 2, 3, 4+)
  - Y-axis: Count of parent coils
  - Data labels: Count and percentage
  - Colors: Gradient

Right (50%):
- Scatter Chart: "Parent Yield Analysis"
  - X-axis: [Pieces per Parent]
  - Y-axis: [Prime Rate %]
  - Size: total_cycle_time_min
  - Color: Product Band
  - Trend line: Show correlation

Row 6: Detailed Product Table

- Table: "Product Performance Detail"
  - Rows: type_code
  - Columns:
    - [Total Pieces]
    - [Prime Rate %]
    - [Avg Cycle Time]
    - [Pieces per Hour]
    - [Avg Completion Gap]
  - Subtotals: By Product Category
  - Conditional formatting: All numeric columns

Filters:
- Date Range
- Product Band
- Type Code (multi-select)
- Thickness Range (slider)
- Width Range (slider)
- Shift Code
- Prime/Scrap filter

---

5. Advanced Features

5.1 Drill-Through Pages

Create drill-through pages for detailed analysis:

Equipment Detail Drill-Through:
- Trigger: Right-click any equipment name

- Fields passed: equipment_id, equipment_name
- Content:
  - Equipment specs (from dim_equipment)
  - Time series of operation duration
  - Maintenance history for this equipment
  - Coil processing timeline
  - RUN/IDLE/FAULT breakdown

Parent Coil Drill-Through:
- Trigger: Right-click parent_coil_id
- Fields passed: parent_coil_id
- Content:
  - All pieces from this parent (UID breakdown)
  - Completion timeline
  - Prime vs scrap split
  - Total cycle time
  - Equipment path visualization

5.2 Bookmarks

Create bookmarks for quick navigation:

1. "Current Month View" - Sets date filter to current month
2. "Bottleneck Equipment Only" - Filters to bottleneck equipment
3. "Prime Products Only" - Filters to prime products
4. "Shift A Performance" - Filters to Shift A
5. "Last 7 Days" - Sets rolling 7-day window

5.3 What-If Parameters

Create what-if parameters for scenario analysis:

Tempo Target Parameter:
```dax
Tempo Target Parameter =
GENERATESERIES(2.0, 6.0, 0.5)
```

Monthly Target Parameter:
```dax
Monthly Target Parameter =
GENERATESERIES(5000, 20000, 1000)
```

Planned Downtime Parameter:
```dax
Planned Downtime Hours Parameter =
```

```
GENERATESERIES(0, 100, 5)
```

Use these in measures:
```dax
Target Achievement with Parameter =
DIVIDE(
    [MTD Pieces],
    'Monthly Target Parameter'[Monthly Target Parameter Value],
    0
) * 100
```

5.4 Tooltips

Create custom tooltip pages:

Equipment Tooltip:
- equipment_name
- [Equipment Utilization %]
- [Total RUN Hours]
- [Total FAULT Hours]
- Mini bar chart: Last 7 days utilization

Coil Tooltip:
- coil_id
- parent_coil_id
- type_code
- completion_ts
- total_cycle_time_min
- Prime/Scrap indicator

---

6. Report-Level Settings

6.1 Theme and Branding

Apply custom theme with ArcelorMittal branding:
- Primary Color: #003087 (ArcelorMittal Blue)
- Secondary Color: #E30613 (ArcelorMittal Red)
- Accent Colors: Use for different shift codes
- Font: Calibri or Arial
- Background: White or Light Gray (#F5F5F5)

6.2 Mobile Layout

Create mobile-optimized layouts for each dashboard:
- Simplified KPI cards only on first screen
- Swipeable charts
- Larger touch targets
- Reduced visual complexity
- Priority on key metrics

6.3 Row-Level Security (RLS)

Implement RLS if needed:

```dax
// Shift Supervisors see only their shift
[shift_code] = USERNAME()

// Maintenance only sees maintenance data
[role] = "Maintenance"

// Executives see everything
[role] = "Executive" || [role] = "Admin"
```

6.4 Refresh Schedule

Set up automatic refresh schedule:
- Production data: Every 1 hour during production hours
- Maintenance data: Every 4 hours
- Equipment events: Every 1 hour
- Full refresh: Daily at 2:00 AM

---

7. Performance Optimization

7.1 Query Reduction

- Use aggregation tables for common calculations
- Create summary tables for date intelligence
- Implement incremental refresh for large fact tables

7.2 Measure Optimization

- Use CALCULATE instead of FILTER where possible
- Avoid unnecessary context transitions
- Use variables to store intermediate calculations
- Leverage SELECTEDVALUE for single-value contexts

7.3 Visual Optimization

- Limit visuals per page to 10-15
- Reduce data points in charts (aggregate if >10,000 points)
- Use Top N filtering instead of showing all values
- Enable Show Items with No Data only when necessary

---

8. User Training

8.1 Quick Start Guide

Create a one-page quick reference:
- How to navigate dashboards
- How to use filters and slicers
- How to drill-through
- How to export data
- How to refresh data

8.2 Key Insights to Monitor

Train users to watch for:
1. Tempo falling below target
2. Equipment utilization drops
3. Increased maintenance events
4. Shift handover gaps increasing
5. Prime rate declining
6. Bottleneck equipment changing
7. Product mix shifts

8.3 Action Triggers

Define when to take action:
- Tempo < 3.5 pieces/hour for 3 consecutive days → Investigate bottlenecks
- Equipment downtime > 15% → Schedule maintenance review
- Shift efficiency variance > 20% → Investigate crew training
- Prime rate < 80% → Review quality procedures
- Handover loss > 15 min → Review shift change procedures

---

9. Maintenance and Updates

9.1 Monthly Review

- Verify data accuracy

- Update targets and thresholds
- Review user feedback
- Optimize slow-performing visuals
- Update documentation

9.2 Quarterly Enhancements

- Add new requested metrics
- Refine bottleneck algorithms
- Update predictive models
- Expand drill-through capabilities
- Enhance mobile experience

9.3 Data Quality Checks

Implement automated checks:
- Missing data detection
- Outlier identification
- Timestamp validation
- Relationship integrity
- Measure consistency

---

10. Success Metrics

Track dashboard adoption and impact:

Usage Metrics:
- Daily active users
- Average session duration
- Most viewed dashboards
- Most used filters
- Export frequency

Business Impact:
- Tempo improvement %
- Reduction in manual reporting hours
- Faster issue identification time
- Maintenance cost reduction
- Production target achievement rate

Appendix A: Glossary

Tempo: Rate of production measured in pieces per hour

Cycle Time: Time from coil entry to exit through the temper line
Prime: Saleable product (types HL, HM, 98)
Scrap: Waste or trim pieces (types HX, HY, HZ)
Parent Coil (CID): Original coil that gets processed
Output Piece (UID): Individual pieces produced from parent coil
Bottleneck: Equipment that limits overall line throughput
MTBF: Mean Time Between Failures
MTTR: Mean Time To Repair
Fast Band: Thin (<2mm) and narrow (<1300mm) products that process faster

---

Appendix B: Data Dictionary

See separate DAX Formulas document for complete measure definitions.

Appendix C: Troubleshooting

Common Issues:

1. **Slow dashboard loading:**
   - Check for circular relationships
   - Reduce number of visuals
   - Optimize DAX measures
   - Implement aggregations

2. **Incorrect totals:**
   - Verify measure context
   - Check for missing CALCULATE
   - Review relationship directions
   - Validate filter context

3. **Missing data:**
   - Verify CSV import success
   - Check relationship cardinality
   - Review filter settings
   - Validate data types

4. **Performance degradation:**
   - Clear cache
   - Refresh data model
   - Check gateway connection
   - Review query folding

Document Control

Version: 1.0
Date: December 2024
Author: Process Engineering Team
Classification: Internal Use
Next Review: March 2025

END OF POWER BI IMPLEMENTATION GUIDE