Business Performance Monitoring Tracker - Complete Excel Solution

Overview

This solution creates a comprehensive Excel tracker with 4 main tabs plus supporting data sheets, all with automated calculations and macros for easy data entry and reporting.

Workbook Structure

Tab 1: Team Performance Dashboard

Purpose: Track individual team members grouped by Line of Business **Key Metrics**:

- Availability = Non-Productive Time (Hours) / [Testing Time Available (Productive Time) - PTO Hours]
- Availability %
- Total Productive Time (Hours of Productivity)

Tab 2: Weekly Business Line Performance

Purpose: Track 4 Lines of Business weekly performance **Key Metrics**:

- Predictability = Completed Tests / Planned Tests
- Predictability %
- Productivity = Test Cases Completed per Day / Test Cases Assigned (Daily Quota)
- Productivity %

Tab 3: Raw Productivity Data

Purpose: Input sheet for daily productivity tracking

Tab 4: Approved Time Off

Purpose: PTO tracking for availability calculations

Tab 5: Team Configuration

Purpose: Setup sheet for team members, quotas, and business line assignments

---

Excel Setup Instructions

Step 1: Create the Workbook Structure

1. Open Excel and create a new workbook
2. Rename Sheet1 to "Team Performance"
3. Add these additional sheets:
   - "Business Line Weekly"
   - "Raw Productivity Data"
   - "Approved Time Off"
   - "Team Config"

Step 2: Set Up Team Configuration Sheet

Column Headers (Row 1):

- A1: Employee Name
- B1: Line of Business
- C1: Daily Test Quota
- D1: Full Capacity Per Week
- E1: Reporting Manager

Sample Data (Rows 2-20):

```
Maha          | Medicaid   | 30 | 65.0 | Manager1
Karthick      | Medicaid   | 25 | 20.9 | Manager1
Tremaine      | Medicaid   | 35 | 59.0 | Manager1
Tracy         | Medicaid   | 40 | 80.0 | Manager2
```

Step 3: Set Up Raw Productivity Data Sheet

Column Headers (Row 1):

- A1: Date
- B1: Employee Name
- C1: Line of Business
- D1: Audit Time (Hours)
- E1: Productive Time (Hours)
- F1: Non-Productive Time (Hours)
- G1: Total Hours
- H1: Tests Completed
- I1: Tests Planned
- J1: PTO Hours
- K1: Comments

Step 4: Set Up Approved Time Off Sheet

Column Headers (Row 1):

- A1: Employee Name
- B1: Date
- C1: Line of Business
- D1: Hours
- E1: Type (PTO/Sick/Holiday)
- F1: Comments

Step 5: Create Team Performance Dashboard

Setup the header structure:

- A1: "Team Performance Dashboard"
- A2: Current week date range
- Row 4: Column headers matching your first image

Key Formulas for Team Performance:

Availability Calculation (Column for Availability %):

```excel
=IF(AND(ISNUMBER(E2),ISNUMBER(F2),ISNUMBER(J2)),
   (F2/(E2-J2))*100,
   "N/A")
```

Total Productive Time:

```excel
=SUMIFS('Raw Productivity Data'!E:E,
        'Raw Productivity Data'!B:B,A2,
        'Raw Productivity Data'!A:A,">="&DATE(YEAR(TODAY()),MONTH(TODAY()),DAY(TODAY())-7))
```

Step 6: Create Business Line Weekly Dashboard

Column Headers:

- A1: Line of Business
- B1: Week Ending
- C1: Tests Completed
- D1: Tests Planned
- E1: Predictability %
- F1: Daily Avg Completed
- G1: Daily Quota Total
- H1: Productivity %

Key Formulas for Business Line Performance:

Predictability %:

```excel
=IF(D2>0,(C2/D2)*100,0)
```

Productivity %:

```excel
=IF(G2>0,(F2/G2)*100,0)
```

VBA Macros Code

Macro 1: Update Dashboard Data

vba

```vba
Sub UpdateDashboard()
    Application.ScreenUpdating = False

    ' Update Team Performance calculations
    Call UpdateTeamPerformance

    ' Update Business Line calculations
    Call UpdateBusinessLineData

    ' Refresh pivot tables if any
    ActiveWorkbook.RefreshAll

    Application.ScreenUpdating = True
    MsgBox "Dashboard updated successfully!"
End Sub

Sub UpdateTeamPerformance()
    Dim wsTeam As Worksheet
    Dim wsRaw As Worksheet
    Dim wsConfig As Worksheet
    Dim wsPTO As Worksheet

    Set wsTeam = Worksheets("Team Performance")
    Set wsRaw = Worksheets("Raw Productivity Data")
    Set wsConfig = Worksheets("Team Config")
    Set wsPTO = Worksheets("Approved Time Off")

    ' Clear existing data (keep headers)
    wsTeam.Range("A5:Z100").Clear

    ' Get current week date range
    Dim weekStart As Date
    Dim weekEnd As Date
    weekStart = Date - Weekday(Date, 2) + 1  ' Monday
    weekEnd = weekStart + 6  ' Sunday

    ' Update week header
    wsTeam.Range("A2").Value = "Week: " & Format(weekStart, "mm/dd/yyyy") & " - " & Format(weekEnd, "mm/d

    ' Copy team members from config
    Dim lastRow As Long
    lastRow = wsConfig.Cells(wsConfig.Rows.Count, "A").End(xlUp).Row

    If lastRow > 1 Then
        wsConfig.Range("A2:E" & lastRow).Copy
        wsTeam.Range("A5").PasteSpecial xlPasteValues
        Application.CutCopyMode = False
    End If

    ' Calculate metrics for each team member
    Dim i As Long
    For i = 5 To wsTeam.Cells(wsTeam.Rows.Count, "A").End(xlUp).Row
        If wsTeam.Cells(i, 1).Value <> "" Then
            Call CalculateTeamMetrics(wsTeam, wsRaw, wsPTO, i, weekStart, weekEnd)
        End If
    Next i
End Sub

Sub CalculateTeamMetrics(wsTeam As Worksheet, wsRaw As Worksheet, wsPTO As Worksheet, rowNum As Long
    Dim employeeName As String
    Dim productiveHours As Double
    Dim nonProductiveHours As Double
```

```
    Dim nonProductiveHours As Double
    Dim ptoHours As Double
    Dim testsCompleted As Long
    Dim employeeStatus As String

    employeeName = wsTeam.Cells(rowNum, 1).Value

    ' Calculate totals for the week
    Dim lastRawRow As Long
    lastRawRow = wsRaw.Cells(wsRaw.Rows.Count, "A").End(xlUp).Row

    For j = 2 To lastRawRow
        If wsRaw.Cells(j, 2).Value = employeeName And _
           wsRaw.Cells(j, 1).Value >= weekStart And _
           wsRaw.Cells(j, 1).Value <= weekEnd Then

            productiveHours = productiveHours + wsRaw.Cells(j, 5).Value
            nonProductiveHours = nonProductiveHours + wsRaw.Cells(j, 6).Value
            testsCompleted = testsCompleted + wsRaw.Cells(j, 8).Value
        End If
    Next j

    ' Calculate PTO hours
    Dim lastPTORow As Long
    lastPTORow = wsPTO.Cells(wsPTO.Rows.Count, "A").End(xlUp).Row

    For j = 2 To lastPTORow
        If wsPTO.Cells(j, 1).Value = employeeName And _
           wsPTO.Cells(j, 2).Value >= weekStart And _
           wsPTO.Cells(j, 2).Value <= weekEnd Then
            ptoHours = ptoHours + wsPTO.Cells(j, 4).Value
        End If
    Next j

    ' Determine employee status
    If productiveHours > 0 Then
        employeeStatus = Format((testsCompleted / wsTeam.Cells(rowNum, 3).Value) * 100, "0") & "%"
    Else
        employeeStatus = "#DIV/0!"
    End If

    ' Update the row with calculated values
    wsTeam.Cells(rowNum, 6).Value = productiveHours ' Productive Time
    wsTeam.Cells(rowNum, 7).Value = nonProductiveHours ' Non-Productive Time
    wsTeam.Cells(rowNum, 8).Value = ptoHours ' PTO Hours
    wsTeam.Cells(rowNum, 9).Value = testsCompleted ' Tests Completed
    wsTeam.Cells(rowNum, 10).Value = employeeStatus ' Employee Status

    ' Calculate Availability %
    If (productiveHours - ptoHours) > 0 Then
        wsTeam.Cells(rowNum, 11).Value = Format((nonProductiveHours / (productiveHours - ptoHours)) * 100, "0.0
    Else
        wsTeam.Cells(rowNum, 11).Value = "N/A"
    End If
End Sub

Sub UpdateBusinessLineData()
    Dim wsBL As Worksheet
    Dim wsRaw As Worksheet

    Set wsBL = Worksheets("Business Line Weekly")
    Set wsRaw = Worksheets("Raw Productivity Data")

    ' Clear existing data
```

```
    wsBL.Range("A3:H20").Clear

    ' Get current week
    Dim weekStart As Date
    Dim weekEnd As Date
    weekStart = Date - Weekday(Date, 2) + 1
    weekEnd = weekStart + 6

    ' Business lines to track
    Dim businessLines As Variant
    businessLines = Array("Medicaid", "Appeals", "EQRO", "UM")

    Dim i As Long
    For i = 0 To UBound(businessLines)
        Call CalculateBusinessLineMetrics(wsBL, wsRaw, i + 3, businessLines(i), weekStart, weekEnd)
    Next i
End Sub

Sub CalculateBusinessLineMetrics(wsBL As Worksheet, wsRaw As Worksheet, rowNum As Long, lineName As Stri
    Dim testsCompleted As Long
    Dim testsPlanned As Long
    Dim dailyAvgCompleted As Double
    Dim dailyQuotaTotal As Long

    ' Calculate metrics from raw data
    Dim lastRow As Long
    lastRow = wsRaw.Cells(wsRaw.Rows.Count, "A").End(xlUp).Row

    For j = 2 To lastRow
        If wsRaw.Cells(j, 3).Value = lineName And _
           wsRaw.Cells(j, 1).Value >= weekStart And _
           wsRaw.Cells(j, 1).Value <= weekEnd Then

            testsCompleted = testsCompleted + wsRaw.Cells(j, 8).Value
            testsPlanned = testsPlanned + wsRaw.Cells(j, 9).Value
        End If
    Next j

    dailyAvgCompleted = testsCompleted / 5 ' 5 work days

    ' Get daily quota from config
    Dim wsConfig As Worksheet
    Set wsConfig = Worksheets("Team Config")
    Dim configLastRow As Long
    configLastRow = wsConfig.Cells(wsConfig.Rows.Count, "A").End(xlUp).Row

    For j = 2 To configLastRow
        If wsConfig.Cells(j, 2).Value = lineName Then
            dailyQuotaTotal = dailyQuotaTotal + wsConfig.Cells(j, 3).Value
        End If
    Next j

    ' Update business line row
    wsBL.Cells(rowNum, 1).Value = lineName
    wsBL.Cells(rowNum, 2).Value = Format(weekEnd, "mm/dd/yyyy")
    wsBL.Cells(rowNum, 3).Value = testsCompleted
    wsBL.Cells(rowNum, 4).Value = testsPlanned

    If testsPlanned > 0 Then
        wsBL.Cells(rowNum, 5).Value = Format((testsCompleted / testsPlanned) * 100, "0.0") & "%"
    Else
        wsBL.Cells(rowNum, 5).Value = "N/A"
    End If
```

```
    wsBL.Cells(rowNum, 6).Value = Format(dailyAvgCompleted, "0.0")
    wsBL.Cells(rowNum, 7).Value = dailyQuotaTotal

    If dailyQuotaTotal > 0 Then
        wsBL.Cells(rowNum, 8).Value = Format((dailyAvgCompleted / dailyQuotaTotal) * 100, "0.0") & "%"
    Else
        wsBL.Cells(rowNum, 8).Value = "N/A"
    End If
End Sub
```

Macro 2: Quick Data Entry Form

vba

```
Sub ShowDataEntryForm()
    ' Create a simple input form for quick data entry
    Dim employeeName As String
    Dim workDate As Date
    Dim productiveHours As Double
    Dim testsCompleted As Long

    employeeName = InputBox("Enter Employee Name:", "Data Entry")
    If employeeName = "" Then Exit Sub

    workDate = DateValue(InputBox("Enter Date (mm/dd/yyyy):", "Data Entry", Date))
    productiveHours = Val(InputBox("Enter Productive Hours:", "Data Entry", "8"))
    testsCompleted = Val(InputBox("Enter Tests Completed:", "Data Entry", "0"))

    ' Add to raw data sheet
    Dim wsRaw As Worksheet
    Set wsRaw = Worksheets("Raw Productivity Data")
    Dim newRow As Long
    newRow = wsRaw.Cells(wsRaw.Rows.Count, "A").End(xlUp).Row + 1

    wsRaw.Cells(newRow, 1).Value = workDate
    wsRaw.Cells(newRow, 2).Value = employeeName
    wsRaw.Cells(newRow, 5).Value = productiveHours
    wsRaw.Cells(newRow, 8).Value = testsCompleted

    MsgBox "Data added successfully!"
End Sub
```

Implementation Steps

1. **Create the workbook structure** as outlined above
2. **Set up the basic data sheets** (Team Config, Raw Productivity Data, Approved Time Off)
3. **Enter your team configuration data** in the Team Config sheet
4. **Add the VBA macros** by pressing Alt+F11 and pasting the code into a new module
5. **Create buttons** on your dashboard sheets to run the macros
6. **Test with sample data** to ensure calculations work correctly
7. **Format the sheets** to match your preferred styling

Usage Instructions

1. **Daily**: Enter productivity data using the Quick Data Entry macro or directly in the Raw Productivity Data sheet
2. **Weekly**: Run the Update Dashboard macro to refresh all calculations
3. **As needed**: Add PTO entries to the Approved Time Off sheet
4. **Monthly**: Review and archive old data as needed

This solution provides automated calculations, easy data entry, and comprehensive reporting matching your existing workflow patterns.