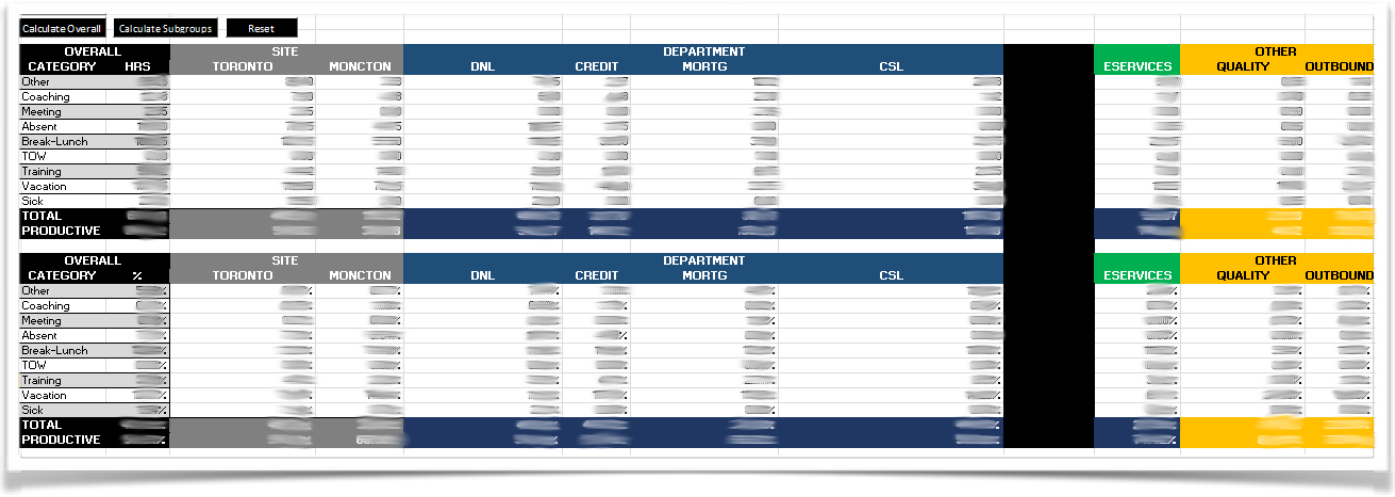
Thomas Eapen
Call Centre Shrinkage Report
Tangerine Bank

Tangerine Call Centre Shrinkage Report



Calculate Overall | Calculate Subgroups | Reset

OVERALL CATEGORY	HRS	SITE TORONTO	SITE MONCTON	DEPARTMENT DNL	DEPARTMENT CREDIT	DEPARTMENT MORTG	DEPARTMENT CSL	OTHER ESERVICES	OTHER QUALITY	OTHER OUTBOUND
Other										
Coaching										
Meeting										
Absent										
Break-Lunch										
TOW										
Training										
Vacation										
Sick										
TOTAL PRODUCTIVE										

OVERALL CATEGORY	%	SITE TORONTO	SITE MONCTON	DEPARTMENT DNL	DEPARTMENT CREDIT	DEPARTMENT MORTG	DEPARTMENT CSL	OTHER ESERVICES	OTHER QUALITY	OTHER OUTBOUND
Other										
Coaching										
Meeting										
Absent										
Break-Lunch										
TOW										
Training										
Vacation										
Sick										
TOTAL PRODUCTIVE										

Summary:

In December 2015, The Tangerine call centre began the “My Evolution” (MEVO) phase and moved from the Aspect WFM platform to the Genesys WFM platform. At this point in time, a lot of the reporting frameworks that were used by the IDA and Workforce teams became invalid due to the incompatibilities in the raw data structures and workflow processes of the two systems.

In the first half of 2016, I began work on building the Shrinkage Report which would become a crucial portion of the reporting frameworks that are used by the Workforce Team. The main feature of the report was that we could calculate shrinkage hours and therefore actual scheduled hours at a 15 minute resolution. In addition, shrinkage calculations took into effect the sub groupings of our call centre staff that matched our channels, site, language and skilling models that were being used by the forecasting and scheduling teams. This fine grained level of insight allows a whole new level of descriptive and predictive analysis to build upon

Features

A summary of the functionality of the Shrinkage Report was mentioned in the above section. Here we go into further detail on the features available in the Tangerine Call Centre Shrinkage Report:

- Calculation of Shrinkage Hours and therefore Actual Scheduled Hours at a 15 minute resolution
- Shrinkage calculations that take into effect the sub groupings of our call centre staff and hence matched to our channel, site, language and skilling models
- Ability to leverage static group information administered by Workforce rather than the Dynamic activity based capability information maintained by Genesys. This allowed us to compensate for potential errors in the activities generated by Genesys.
- Timeframes that ranged from midnight to midnight and accurate handling of timeframes that cross day boundaries (e.g. overnight staff)
- Granular shrinkage buckets and a grouping of shrinkage types allow us to drill down from higher level groupings when needed
- A snapshot feature where we generate actual shrinkage and scheduled hours at the start of the day and are able to compare that snapshot to updated shrinkage and scheduled hours throughout the day
- A “Settings” feature where we are able to add/remove/modify grouping information and shrinkage bucket information through mapping tables and not through code.
- Easy to use with the ability to correct user errors with relatively little effort
- Ability to consolidate Shrinkage into daily, weekly, monthly, annual reports at an overall and sub grouping level with comparisons of forecasted/actual shrinkage.
- High performance where large datasets can be processed in relatively short periods of time

Evolution

The first step in building the Shrinkage sheets involved Sanitizing the Raw Data that is to be summarized by Excel. It was decided early on that a general purpose programming language was needed rather than Excel’s inbuilt formulae. Hence, I decided to use the VBA language that was well integrated in Microsoft Excel. The sanitizing step proved to one of the most crucial where a single run through of all the data checks to see if the data is relevant or not. If the row was not relevant, it was marked as such which allowed us to filter more effectively. If the row of data was relevant, the interested date was transformed into a format that made it compatible for quick sorting and filtering. This allows better accuracy and reduces the amount of processing code needed in the following run throughs of the data. The sanitizing step also increases the performance of the data processing due to much more efficient filtering/sorting capabilities with clear sorting/filtering markers.

The core function that builds the shrinkage sheets is the timeframe processing code where a start and end time are compared to a specific timeframe and the function returns the number of minutes that fall within those specified timeframes. So if an Associate works from 09:00 to 17:00, 1 hour of his schedule would fall in the 07:00 - 10:00 bucket, 4 hours of his schedule would fall in the 10:00 - 14:00 bucket and 3 hours of his schedule would fall in the 14:00 - 18:00 bucket. This core function tends to drive the entire

data pipeline from the sanitizing steps before the function is called to the design steps after the results are calculated. The heart of shrinkage is below.

```
If tCheckEnd <> startOfDay Or tCheckStart <> startOfDay Then
    tStartVal = row.Columns(row.ListObject.ListColumns("Start Time Sanitized").Index).Value
    tEndVal = row.Columns(row.ListObject.ListColumns("End Time Sanitized").Index).Value

    If tEndVal = startOfDay And tStartVal <> startOfDay Then
        tEndVal = midnight
    End If
    If tEndVal > startBoundary Then
        If tStartVal < endBoundary Then
            If tStartVal <= startBoundary Then
                If tEndVal >= endBoundary Then
                    timeCount = timeCount + (endBoundary - startBoundary)
                    instanceCount = instanceCount + 1
                Else
                    timeCount = timeCount + (tEndVal - startBoundary)
                    instanceCount = instanceCount + 1
                End If
                Else
                If tEndVal >= endBoundary Then
                    timeCount = timeCount + (endBoundary - tStartVal)
                    instanceCount = instanceCount + 1
                Else
                    timeCount = timeCount + (tEndVal - tStartVal)
                    instanceCount = instanceCount + 1
                End If
            End If
        End If
    End If
```

After the first iteration of the Shrinkage sheets were built and the accuracy of the results were verified, the performance of the data pipeline was immediately noticed as a problem. One day's worth of schedule information took nearly 30 mins to process on a group level at a 60 minute resolution. The code was re-looked at again from top to bottom to identify the portions of the code that was causing the performance issues. Quite soon, it became apparent that the manual sorting/filtering of the data was causing the performance issues. Raw sanitized data was being stored in Excel tables but the sorting/ filtering portion of the pipeline involved parsing multiple iterations of the same data which was causing unnecessary bottlenecks.

The next step to reduce this bottleneck was to look at the inbuilt features of the "Range" data structure to do the

```
Set wsMappings = ActiveWorkbook.Worksheets(mappingsSheetName)
Set ws = ActiveSheet

 With wsMappings.ListObjects(teamTable)
    teamArray = Application.Transpose(.ListColumns(teamGroup).DataBodyRange.Value)
 End With

 With wsMappings.ListObjects(stateTable)
    stateArray = Application.Transpose(.ListColumns(shrinkageState).DataBodyRange.Value)
 End With

With ws.ListObjects(rawDataTable)
    .AutoFilter.ShowAllData
 End With

With ws.ListObjects(rawDataTable)
    .Range.AutoFilter Field:=11, Criteria1:="<>"
    .Range.AutoFilter Field:=3, Criteria1:=teamArray, Operator:=xlFilterValues
    .Range.AutoFilter Field:=8, Criteria1:=stateArray, Operator:=xlFilterValues
    On Error GoTo ExitFor:
    For Each row In .DataBodyRange.SpecialCells(xlCellTypeVisible).EntireRow
```

heavy lifting portions of the sorting/filtering of data.

The Range structure seen above has multiple AutoFilter options that filter data at a much faster speed than the manual parsing of the data in the first iteration. This speed boost of the 2nd iteration of the Shrinkage Report allowed us to increase the resolution to 15 minute intervals and also contained snapshot information that made the Shrinkage Report a lot more useful than before.

Screenshots to the left and below show us the design of the processed data and their relevance to scheduling efficiencies and performance predictors