THOMAS EAPEN · SELECTED WORK

Follow offline time through the report

Category, group and interval views that reconcile to one synthetic schedule, with the change from snapshot to actual made explicit.

Original tools: **Excel / VBA** · Example calculations: **Python**

Synthetic example · independently invented data · One-page A3 landscape

8
invented members
Two generic groups · four intervals

140
snapshot offline minutes
Person-minutes across the window

175
actual offline minutes
Person-minutes across the window

+35
change from the snapshot
Shared by the downstream dashboard

01 / GROUP AND CATEGORY VIEWS

Keep the same categories in each view

Category	Group A		Group B		Combined change
	Snapshot	Actual	Snapshot	Actual	
Learning	30	35	40	40	+5
Coordination	15	20	25	30	+10
Other	15	30	15	20	+20
All categories	**60**	**85**	**80**	**90**	+35

Offline person-minutes · the same events grouped two ways

Every member is assigned to one group. Offline events for the same member do not overlap; contributions from different members are added.

02 / INTERVAL VIEW

Reconcile each reporting window

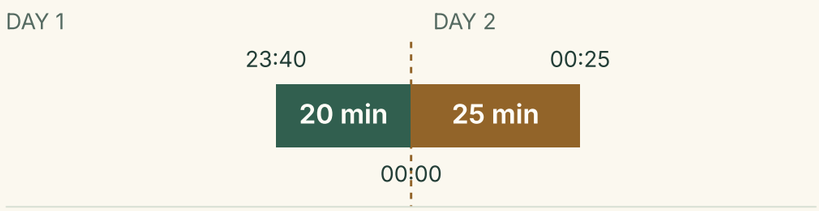
Interval	Snapshot offline	Actual offline	Change	Actual available
09:00–09:30	35	40	+5	200
09:30–10:00	30	50	+20	190
10:00–10:30	40	45	+5	195
10:30–11:00	35	40	+5	200
Window total	**140**	**175**	**+35**	**785**

Invented day · all quantities are person-minutes

Each half-hour contains 240 scheduled person-minutes (8 × 30). Actual available time is scheduled time minus actual offline time: 960 − 175 = 785 minutes.

03 / A SEPARATE BOUNDARY EXAMPLE

Keep midnight inside the calculation



20 + 25 = 45 minutes · keep the day offset in the data

The event has endpoints 1,420 and 1,465 minutes from Day 1 midnight. Its day offset is retained. It is separate from the daytime report above.

SELECTED DEMONSTRATION CODE / PYTHON

Intersect the event and reporting interval

The endpoints are integer minutes on one continuous timeline. Each contribution is reused in the category and interval views.

```
left = max(start, window_start)
right = min(end, window_end)
minutes = max(0, right - left)
```

Selected code from the new calculation that produces this example's results.

One calculation, several useful views. Preparing valid records and retaining reusable interval contributions makes the grouping, comparison and downstream reporting inspectable. See the dashboard that uses these figures ↗

Thomas Eapen
nowlabs.net/tangerine-ida

New illustrative material explaining the approach. Values, configurations and results shown here are not historical workplace records.