

Stop guessing line items. Model *the arithmetic of the business*.

A line-item plan asks eight hundred questions nobody can answer — what will travel be in month nine? A driver-based plan asks thirty that somebody can: how many people, at what cost, hired when; how many customers, at what price, churning at what rate. This article sets out the mathematics of driver-based modeling in plain terms — unit economics, step-function costs, and the breakback arithmetic that reconciles top-down targets with bottom-up drivers — and shows where each piece lives in EAConnect Planning.

AUDIENCE

Leaders of mid-size enterprises

CORE IDENTITY

Amount =
quantity $\times$ rate,
at a grain

THREE
MECHANISMS

Drivers · steps ·
breakback

READING TIME

13 min read

Series Hours, Not Months

Evidence EAConnect planning-entry screen and model navigation, Sept 2026

Standard methodology as practitioner opinion; worked figures are illustrative

01 Why line items fail and drivers do not

A number typed into a cell has no reason attached. A number computed from

a driver carries its reason with it.

The line-item plan is the spreadsheet everyone has built: rows of GL accounts, columns of months, and a value in every cell that somebody estimated, copied from last year, or grew by a percentage. It has two structural problems. It cannot explain itself — when marketing comes in 26% under, the plan offers no decomposition of why — and it cannot be changed coherently. Move the hiring plan by two months and forty cells across salaries, benefits, payroll tax, equipment and software should all move together. In a line-item plan they do not, because nothing links them.

A driver-based plan inverts the structure. A small set of operational quantities — headcount by role, customers by segment, volume by product, square footage — are the inputs. Everything in the P&L is derived from them by rate and by rule. Change the hiring plan, and every dependent line recomputes. Ask why marketing is under, and the answer is a driver: campaigns ran, cost per lead, leads generated.

The test of a driver

A driver is a quantity that a named person can defend, that moves the plan materially when it changes, and that nobody has an incentive to misstate. Headcount plan by month passes all three. "Other opex as 4% of revenue" passes none — it is a rate masquerading as a driver.

02 The anatomy: quantity, rate, grain, time

Every planned amount is a quantity times a rate, computed at a specific grain, in a specific period. Getting those four things explicit is most of the work.

```
Amount[line, entity, cost centre, period]
    = Quantity[driver, entity, cost centre, period] × Rate[driver → li
```

Three kinds of driver do almost all the work in a mid-size company's plan:

QUANTITY DRIVERS

How many

Headcount by role and level. Customers by segment. Units by product. Square feet by site. Active projects. These are counts a manager owns.

RATE DRIVERS

At what price

Average compensation by band. Price per unit. Cost per lead. Rent per square foot. Benefits load as a percentage of base. Rates are assumptions, set centrally, versioned.

TIMING DRIVERS

When

Hire start dates and ramp. Contract start and renewal dates. Collection days. Capex in-service dates. Timing is what turns an annual number into a monthly shape — and a monthly shape into cash.

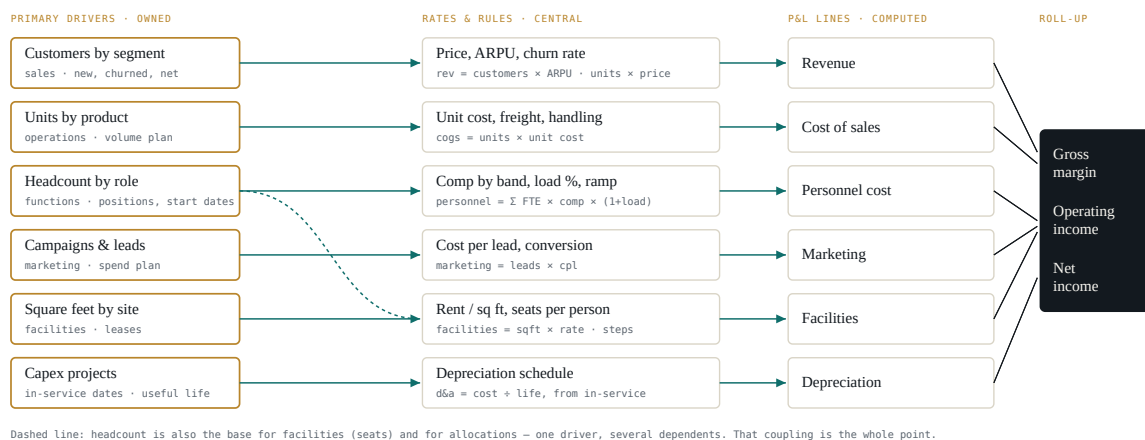


Figure 1. The driver tree of a typical mid-size company. Six owned quantities on the left, central rates and rules in the middle, computed lines on the right. Nothing in the right-hand column is typed.

Grain

The grain of a driver is the combination of dimensions at which it is entered: headcount by *role × level × cost centre × entity × month*, say. The rule that prevents most modeling accidents is that a computed line inherits the grain of its coarsest input. If compensation rates are held by band and country, personnel cost cannot be more granular than band and country without an allocation. Deciding grain early — and writing it down on each driver — is the single most valuable hour in a model design.

03 A worked example: revenue and personnel

Two lines, fully specified, with the numbers carried through. Both use only the identity in section 02; the interest is in the timing terms.

Subscription revenue

```
Customers[t] = Customers[t-1] + New[t] - Churned[t]
Churned[t]   = Customers[t-1] × churn_rate[t]
Revenue[t]   = ( Customers[t-1] + Customers[t] ) / 2 × ARPU[t] ←
```

MONTH	OPENING	NEW	CHURN 2.0%	CLOSING	AVG	ARPU	REVENUE
M1	1,000	60	20	1,040	1,020	180	183,600
M2	1,040	65	21	1,084	1,062	180	191,160
M3	1,084	70	22	1,132	1,108	182	201,656
Quarter		195	63	1,132			576,416

Three owned drivers (opening base, new customers, churn rate), one central rate (ARPU), one rule (average count). The sales leader defends "new"; the customer team defends churn; finance owns ARPU. Nobody types revenue.

Personnel cost with hiring ramp

```
FTE[role, t]      = Filled[role, t-1] + Starts[role, t] × start_fraction
Personnel[role, t] = FTE[role, t] × (Base[band] / 12) × (1 + load%) +
start_fraction    = days remaining in month after start date ÷ days i
```

MONTH	PLAN	STARTS	START FRACTION	FTE (PAID)	BASE/YR	LOAD	SALARY + LOAD
M1	2 engineers, start 15th	2	0.50	1.0	120,000	22%	12,200
M2	—	0	—	2.0	120,000	22%	24,400
M3	1 engineer, start 1st	1	1.00	3.0	120,000	22%	36,600
Quarter		3					73,200
Compare with the flat-multiplier version — $3 \text{ heads} \times 120,000 \times 1.22 \div 12 \times 3 \text{ months} = 109,800$ in salary alone, with the recruiting fees missing entirely and the first-month cash out by half. The timing terms are what make the plan match the bank statement.							

The second table is why headcount planning gets its own article in this library: payroll is the largest opex line in most mid-size companies, and its shape over the year is set almost entirely by start dates, ramp and load — none of which a flat average captures.

04 Step-function costs

Some costs do not scale smoothly with a driver. They jump when a threshold is crossed — a new support hire every 400 customers, a new warehouse at 90% of capacity, a new licence tier at 250 users. Modeling them as a percentage produces a plan that is wrong in every month and right on average.

```
Required[t] = ceil( Driver[t] ÷ capacity_per_unit )
Cost[t]      = max( Required[t], Committed[t] ) × unit_cost[t]
Trigger[t]   = Required[t] > Required[t-1]
```

SUPPORT HEADCOUNT vs CUSTOMERS · ONE AGENT PER 400 CUSTOMERS · ILLUSTRATIVE

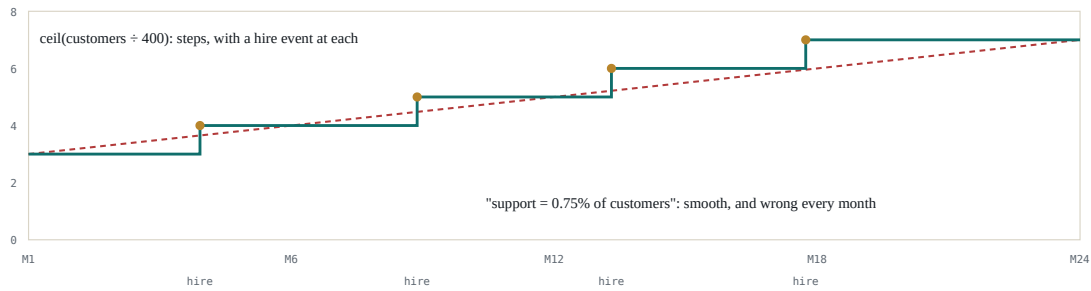


Figure 2. A step cost against its smooth approximation. The gold dots are events — each one has a lead time, a recruiting cost and a month in which the new person is paid but not yet productive. A percentage-of-revenue line hides all of that.

The step function matters for three reasons beyond accuracy. It produces *events* — "we will need to open the search for a fourth support agent in month five" — which is the kind of statement a plan should make and a percentage never can. It carries a *ratchet*: capacity added is rarely removed when the driver dips, and the `max(required, committed)` term captures that. And it exposes *utilisation*: the gap between the step line and the smooth line is idle capacity, which is a legitimate planning choice when made deliberately and an expensive accident when not.

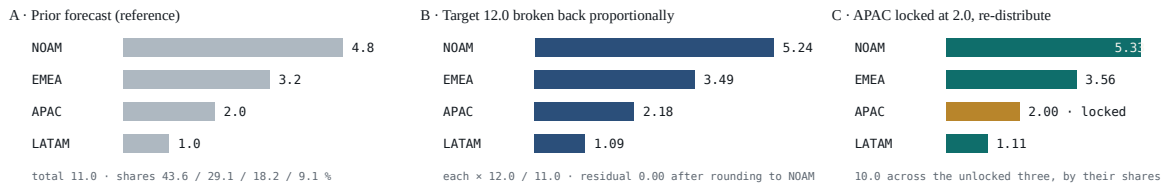
Common step costs in a mid-size company: customer-facing headcount (support, onboarding, account management), warehouse and fulfilment capacity, software licence tiers, server and cloud reservations, office seats and leases, and audit or compliance thresholds triggered by revenue or headcount bands.

05 Breakback: reconciling top-down and bottom-up

Leadership sets a total. Functions build drivers. The two never agree on the first pass, and the arithmetic that reconciles them is called breakback — distributing a parent value down to its children in proportion to a reference, while respecting anything that has been locked.

```
Child_new[i] = Parent_new × ( Ref[i] ÷ Σ Ref[j] )
with locked children L:
Child_new[i] = ( Parent_new - Σ Child[L] ) × ( Ref[i] ÷ Σ Ref[j ∉ L] )
rounding:      assign the residual after rounding to the largest child,
```

BREAKBACK OF A REVENUE TARGET · 4 REGIONS · REFERENCE = PRIOR FORECAST · ILLUSTRATIVE, \$M



The reference decides everything. Prior forecast keeps the shape; actuals-to-date bias toward what happened; driver weight (e.g. sales headcount) lets the target follow capacity. Locks are the negotiation: a region that has committed a number keeps it; the remainder absorbs the change. Rounding residuals go to the largest child so the parent is exact.

The same arithmetic runs down every hierarchy: entity → cost centre → account → month. A target can be broken back three levels in one operation.

Figure 3. Breakback in three steps. The target moves; the shape is preserved; a locked child holds and the others absorb. This is how a top-down number and a bottom-up build meet without either side re-keying the other's work.

Choosing the reference

Breakback is only as sensible as its reference. Three choices, each with a use: the *prior version* preserves the shape of the existing plan and is the right default for a target change mid-cycle; *actuals to date* biases the distribution toward demonstrated performance and is appropriate for in-year re-forecasts; a *driver weight* such as sales headcount or installed base lets the target follow capacity, which is the honest choice when the target is aspirational. A plan should record which reference was used for each breakback, because the answer to "why is EMEA's target 3.56 and not 3.49" is "because APAC was locked and the remainder was distributed by prior share", and that should not have to be reconstructed from memory.

Time breakback

The same arithmetic distributes an annual number across months. The reference is a seasonality profile — last year's monthly shape, or a driver's monthly plan — and the result is a monthly line that sums exactly to the year. A common failure in planning tools is time breakback that spreads evenly, producing a flat line nobody believes; the profile is what makes the monthly plan usable for cash.

06 Allocation: the other direction

Breakback pushes a target down a hierarchy. Allocation pushes a shared cost across a base. They look similar and are governed by different rules, and

confusing them is the source of most "why did my cost centre change" complaints.

$$\text{Allocated}[\text{to}, t] = \text{Pool}[t] \times (\text{Base}[\text{to}, t] \div \sum \text{Base}[\text{all}, t]) \quad \leftarrow$$

two-step: Pool $\rightarrow$ intermediate (e.g. IT to functions by headcount) $\rightarrow$ f

SHARED COST	SENSIBLE BASE	WHY	POOR BASE
Facilities, rent	Seats occupied or square feet	Cost is driven by space	Revenue — a growing team in a small office is under-charged
IT, software seats	Headcount	Cost is per person	Square feet
Finance, HR, legal	Headcount, or transaction volume where known	Effort scales with people	Revenue — penalises high-margin units for existing
Cloud infrastructure	Metered usage where available; else customers	Cost is driven by consumption	Headcount
Corporate overhead to products	Contribution margin, if the purpose is pricing	Tests whether the product carries its share	Equal split — a plan choice disguised as arithmetic
The base is itself a driver from Figure 1. That is what makes allocation stable: when headcount moves, IT allocations move with it, for a reason anyone can trace.			

Two rules keep allocations honest. First, allocate at the grain of the base and no finer — if headcount is planned by cost centre, IT cannot be allocated by product without a second step and a second base. Second, publish allocated cost separately from direct cost in every report. A manager should be able to see what they control and what they carry; blending the two is how allocation becomes a source of argument rather than information.

07 Formula rules that keep a model sane

A driver model is a small program. These are the conventions that keep it

readable and correct after the people who built it have moved on.

RULE	WHAT IT MEANS	WHAT GOES WRONG WITHOUT IT
One direction of flow	Drivers → rates → lines → roll-ups. Nothing computed feeds back into a driver.	Circular references: personnel cost that depends on revenue that depends on headcount that depends on personnel cost. Tools "solve" these iteratively and the answer is whatever the iteration converged on.
Explicit grain on every measure	Each driver, rate and line states the dimensions it is held at.	Silent aggregation: a rate held by country applied at cost-centre grain, so every cost centre in a country gets the same number and nobody notices.
Units in the name	headcount_fte, comp_annual_usd, churn_rate_monthly_pct.	An annual rate applied monthly; a percentage entered as 4 instead of 0.04. Both are common and both survive review because the label was "rate".
Timing as a first-class term	Start fractions, lags, in-service dates and collection days are inputs, not adjustments made afterward.	A plan whose annual total is right and whose monthly shape is wrong, which means its cash forecast is wrong.
Rates are versioned centrally	Compensation bands, prices, load percentages live in one place with a version and an owner.	Forty copies of the benefits load, thirty-eight of them stale.
Steps are ceilings, with memory	ceil() for capacity; max(required, committed) so capacity does not vanish on a dip.	Support headcount that falls in a slow month and re-hires the next, on paper.
Breakback records its reference	Every distributed number carries which reference and which locks produced it.	Targets nobody can explain, defended by nobody.

Direct and
allocated
kept apart

Two measures, both shown.

Managers held to costs they do
not control, and arguments about
the base instead of the business.

None of these is platform-specific. They are the conventions a good modeler applies in any tool, and the ones a planning platform should make easy rather than possible.

08 Where each piece lives in EAConnect Planning

The model objects in the platform map directly onto the anatomy above. The mapping is stated here as far as the navigation and the entry screen show it; the formula and process editors are the subject of the AI-assisted authoring article.

BUILD · DIMENSIONS

Grain

Account, entity, cost centre, scenario, version and time are dimensions; hierarchies on them are versioned with effective dates. A driver's grain is a choice of these.

BUILD · MODELS & PLANNING TABLES

Drivers, rates, lines

A model is a set of measures at a grain. Rates and drivers can be their own models — headcount, exchange rates, assumptions — publishing into the P&L model rather than cluttering it.

BUILD · VARIABLES

Central rates

Load percentages, capacity-per-unit, cost per lead: held once, versioned, referenced by name from formulas.

BUILD · FORMS & VIEWS

Owned entry

Functional owners enter drivers on forms scoped to their cost centre and entity; views present computed lines back to them. The entry screen below is one such form.

BUILD · PROCESS BUILDER

Calculation order

Publish rates → compute drivers → derive lines → allocate → roll up, as an ordered process with validation between steps — the same engine that runs data loads and approvals.

BUILD · CALENDARS

Time

Start fractions, lags and seasonality profiles depend on the calendar; the time dimension is generated from it, so fiscal and 4-4-5 calendars carry through every timing term.

ACCOUNT	2027-01	2027-02	2027-03	2027-04	2027-05	2027-06
IS_NET_INCOME - Net Income	28,101	92,527	15,352	60,386	16,950	4,798
IS_OPERATING_PROFIT - Operating Profit	28,101	92,527	15,352	60,386	16,950	4,798
IS_GROSS_PROFIT - Gross Profit	28,101	92,527	15,352	60,386	16,950	4,798
4000 - Revenue	-1,103	-2,331	-1,323	-2,547	-924	-2,244
4001 - Equipment Sales Revenue						
4002 - Maintenance Revenue						
4003 - Engineering Services Revenue						
4004 - Other Operating Revenue						
4005 - Non Project Service Revenue						
4006 - Rental Income	-1,103	-2,331	-1,323	-2,547	-924	-2,244
4007 - Interest Income						
4008 - Change Order Revenue						
4009 - Warranty Service Revenue						
4010 - Commissioning Revenue						
4011 - Spare Parts Revenue						
5000 - Cost of Goods Sold	29,204	94,858	16,674	62,933	17,874	7,042

Figure 4. An entry form in EAConnect Planning. Roll-up rows (Net Income, Operating Profit, Gross Profit) are computed; the leaf rows are where drivers or values are entered per month and version. The subsidiary selector scopes the form to the owner's entity.

One observation on the mapping. In spreadsheet-based planning the modeler's skill goes into keeping formulas consistent across hundreds of copies. In a dimensional platform the formula is written once at a grain and applied everywhere the grain exists, so the skill shifts to choosing grains and references well — which is what sections 02 and 05 of this article are about.

09 A practitioner's view for leaders

Opinions from people who have built driver models in SAP BPC, Anaplan, Pigment

and OneStream, and rebuilt them when they went wrong. Not product claims.

Thirty drivers, each with a name on it

A driver without an owner is a line item with extra steps. Fewer, owned drivers beat many unowned ones — and the exercise of assigning owners tells you which quantities the business actually manages.

Timing is where the cash lives

Annual totals are easy to get right and nearly useless. Start dates, ramps, collection days and in-service dates are what make a plan's months match the bank. Insist that timing is modeled, not adjusted.

Steps, not percentages, for capacity

Any cost that comes in people, leases, tiers or servers is a step. Model it as one and the plan will tell you when to hire and when to sign; model it as a percentage and it will tell you nothing.

Breakback is negotiation made explicit

Top-down targets and bottom-up builds are supposed to disagree. Proportional breakback with locks is how the disagreement is resolved transparently — and the reference used is a decision that should be recorded, not a default.

Do not allocate what you cannot trace

Every allocation should follow a base that is itself a planned driver. If the base cannot be named, the allocation is an opinion, and it should be labeled as one.

Choose grain before choosing a tool

The question "at what level do we plan headcount, revenue and cost" is answered by the business, not the platform. Answer it first, write it down, and the model build in any tool becomes a week rather than a quarter.

EACConnect Planning · Research library · Hours, Not Months

Evidence: EACConnect planning-entry screen and Build navigation (Dimensions, Models, Planning Tables, Variables, Forms, Views, Process Builder, Calendars), Sept 2026. Worked figures are illustrative. Methodology is practitioner opinion.

2,963 words · 13 min read