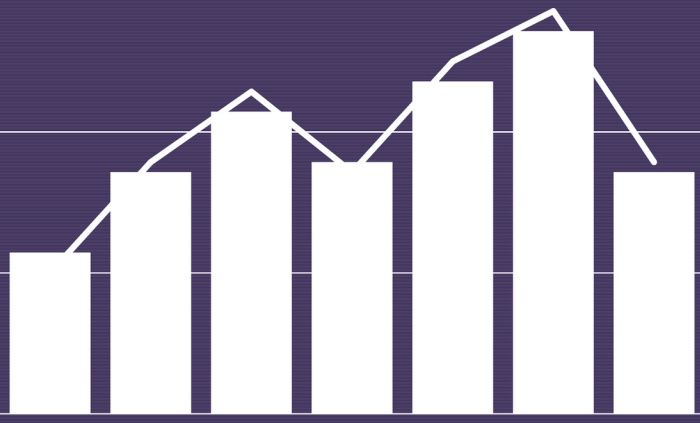
ccplanning.net
workforce planning, done well



A CCPLANNING.NET WHITE PAPER · PART

The Scheduling Masterclass

From the staffing requirement to a roster that covers the curve, fits the people, and holds up on the day.

November 2026 · ccplanning.net

Includes a free schedule coverage spreadsheet

Contents

Right-click and choose "Update Field" to populate page numbers in Word.

Executive summary

Forecasting tells you how many people you need, interval by interval. Scheduling is the craft of actually putting them there — turning a requirement curve into real shifts worked by real people, at the lowest sustainable cost, while hitting service and keeping the workforce willing to stay. It is where a good forecast is either realised or wasted, and it is consistently underrated as a discipline: most operations forecast with care and then schedule with whatever the WFM tool spits out.

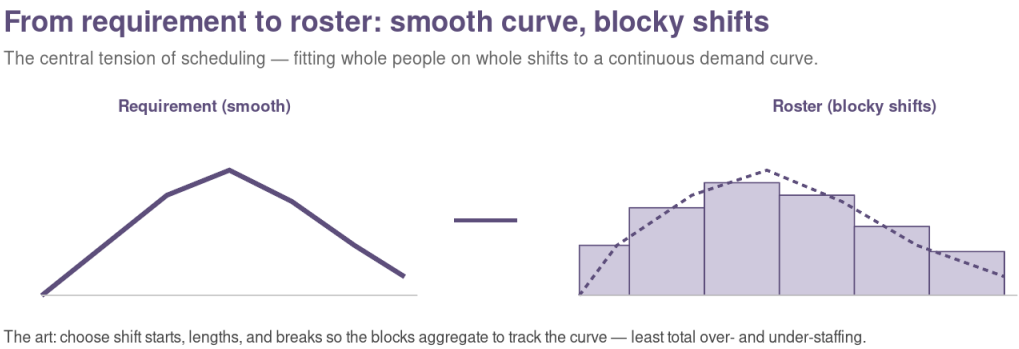
This masterclass walks the scheduling craft in order: turning the requirement into a roster, the sequence to build in, the menu of shift patterns and when each fits, matching shifts to the demand curve, handling shrinkage in the roster, the flexibility levers that close the gaps, designing schedules people will actually adhere to, and measuring coverage honestly. It is written for the planner or scheduler who builds the rosters and wants them to cover better and cost less.

> **The thesis in one paragraph**
>
> A schedule is judged by one thing above all: how well the people on it line up with the demand curve, interval by interval. Everything else — the patterns, the flex layers, the multi-skilling, the fairness rules — is in service of that fit. The schedules that fail do so because they were built to suit the operation's convenience or the spreadsheet's defaults rather than the shape of demand, and because shrinkage and human reality were bolted on at the end instead of designed in from the start. Build to the curve, design in shrinkage and the people, and measure coverage honestly — and the forecast finally turns into service.

This is the fifth ccplanning white paper, and the second of the craft pair. The fourth covered forecasting — producing the requirement. This one covers scheduling — meeting it.

1. From requirement to roster

Scheduling begins where forecasting ends: with a staffing requirement, expressed as a number of agents needed in each interval to hit the service target. That requirement is rarely flat — it rises and falls across the day in a curve, peaks on some weekdays, and shifts with season. The scheduling job is to lay real shifts over that curve so that the people available in each interval match the people needed, as closely as a finite set of humans on finite shift patterns allows.



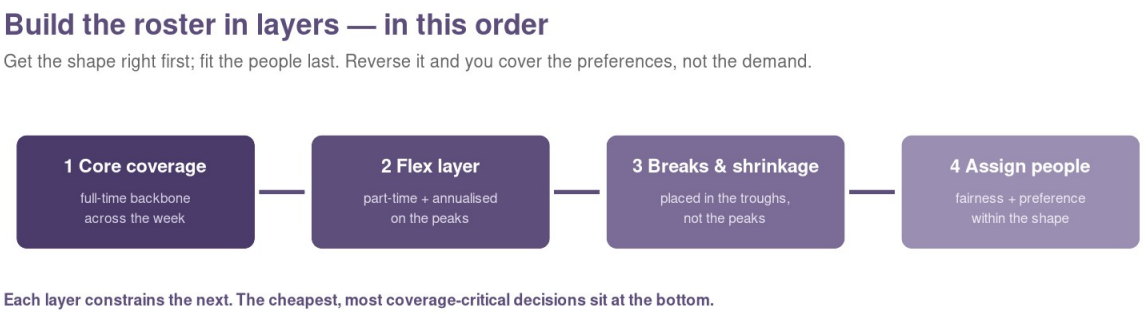
The gap between the smooth requirement curve and the blocky reality of shifts is the central tension of scheduling. You cannot roster 14.3 people from 10:15 to 10:45; you roster whole people on shifts of several hours. The art is in choosing shift start times, lengths, and break placements so the aggregate coverage tracks the curve with the least over- and under-staffing. A schedule that ignores the curve — everyone nine-to-five when demand peaks at 11 and again at 7pm — wastes money in the troughs and misses service at the peaks, no matter how good the forecast behind it was.

It is worth being clear about why the requirement is a curve and not a number, because the shape is what the whole craft responds to. Contact demand follows the rhythm of customers' lives: it builds through the morning as people start their day, often dips around lunch, and in many consumer operations rises again in the early evening as people deal with personal admin after work. Overlaid on that daily shape is the weekly one — Mondays heavier, weekends lighter for most operations — and the seasonal one on top of that. The requirement curve the scheduler receives is the product of all three, and it is rarely gentle. The steeper and more twin-peaked it is, the more the schedule has to work to track it, and the more a flat, single-start roster will cost in wasted hours and missed service. Reading the curve — knowing where the peaks sit and how sharp they are — is the first act of scheduling, before a single shift is placed.

> *A schedule is judged on how well the people on it line up with the demand curve. Everything else is in service of that fit.*

2. The build sequence

Rosters are best built in layers, and the order matters as much as it does in forecasting. Building all at once, or starting with individual preferences before the shape is right, produces a schedule that fits people and misses demand.

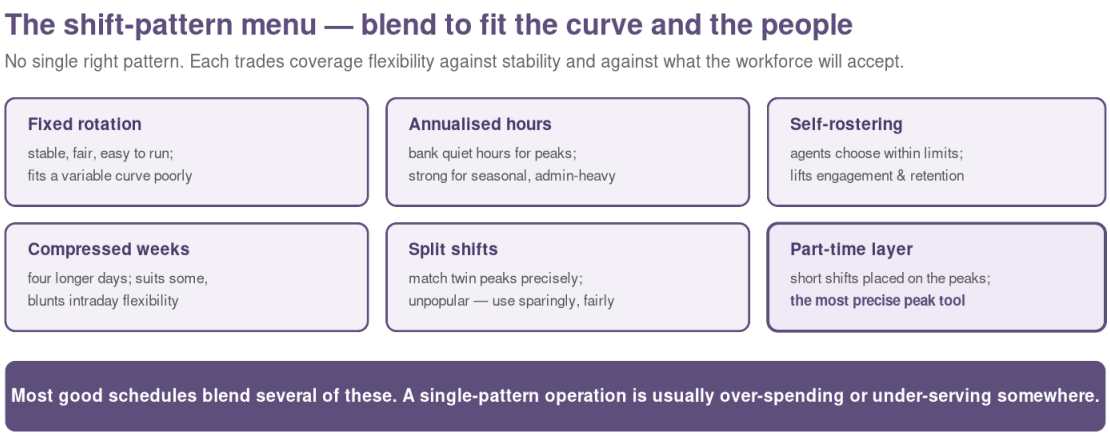


First, lay the **core coverage** — the steady backbone of full-time shifts that covers the bulk of the requirement curve across the week. Second, add the **flex layer** — part-time, annualised hours, and adjustable shifts that fill the peaks the core cannot economically cover. Third, handle **breaks and shrinkage** deliberately, scheduling lunches and breaks into the troughs rather than letting them fall in the peaks. Only then, fourth, comes **individual assignment** — fitting named people to the shifts, honouring preferences and fairness within the shape that already works. Get the shape right first; fit the people second. Reverse the order and you optimise for the roster nobody complains about rather than the one that covers.

The reason the sequence matters is that each layer constrains the next, and the cheapest, most coverage-critical decisions sit at the bottom. Core coverage is the largest cost and the hardest to change, so it should be set against the demand shape with no other consideration intruding. The flex layer is where the curve actually gets tracked, so it must be free to go where the peaks are rather than being squeezed around an already-fixed set of individual preferences. Breaks and planned shrinkage are coverage decisions disguised as administrative ones, so they belong before assignment, not after. And individual assignment — the layer agents care about most — comes last precisely because it is the layer with the most give: there are usually many ways to fit named people to a working shape, but very few ways to fit a working shape around named people. Teams that invert this, starting from who wants what, end up defending a roster that keeps everyone content and misses the evening peak every day.

3. The shift-pattern menu

There is no single right shift pattern; there is a menu, and the craft is choosing the mix that fits the demand shape and the workforce. Each pattern trades coverage flexibility against stability and against what the workforce will accept.

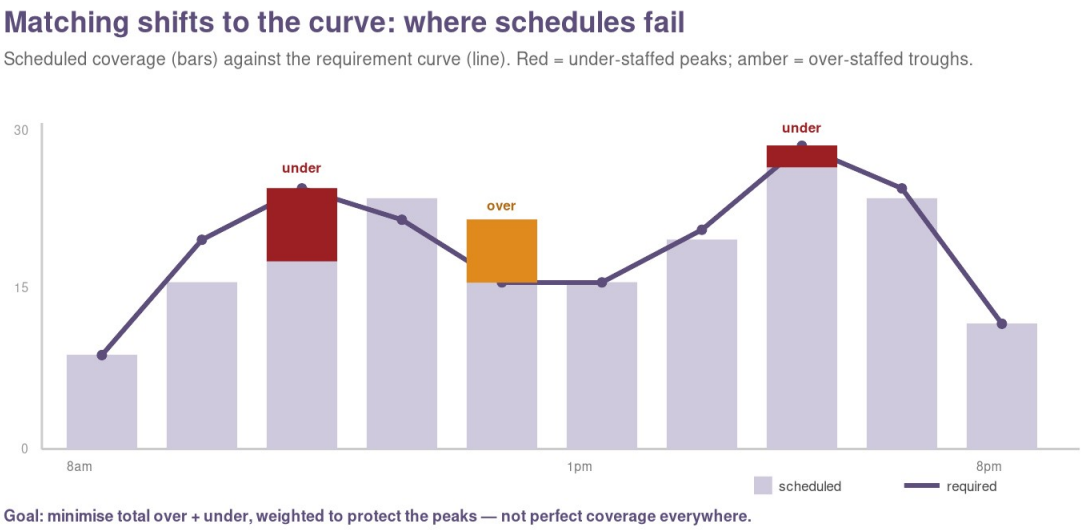


Fixed rotations give stability and fairness and are easy to administer, but fit the curve poorly when demand is variable. **Annualised hours** let you bank hours in quiet periods and spend them in peaks — powerful for seasonal operations, demanding to administer. **Self-rostering** lets agents choose within coverage constraints, which lifts engagement and retention where it can be made to work. **Compressed weeks** (four longer days) suit some lifestyles and reduce changeover but blunt intraday flexibility. **Split shifts** match twin-peak demand precisely but are unpopular and must be used sparingly and fairly. And the **part-time layer** is the single most useful flexibility tool for tracking peaks, because short shifts can be placed exactly where the curve is highest. Most good schedules blend several of these rather than relying on one.

The choice is governed by the demand shape and the workforce, not by fashion. A stable, low-variability operation can run mostly on fixed rotations and lose little; a sharply seasonal one will leave money on the table without annualised hours or a strong flex layer. A twin-peak day — common in consumer operations where people contact before and after the working day — almost demands either split shifts or a staggered mix of part-time and full-time starts, because no single block of full-time shifts covers two separated peaks economically. The workforce matters as much as the demand: a younger, flexibility-seeking workforce may welcome self-rostering and compressed weeks that an older, stability-seeking one would resent. The skill is reading both the curve and the people, and blending patterns to fit — a single-pattern operation is almost always either over-spending or under-serving somewhere.

4. Matching shifts to the curve

This is the heart of the craft: arranging shift starts, lengths, and breaks so aggregate coverage tracks the requirement curve. Two failure shapes recur, and a good schedule avoids both.



The first failure is **over-staffing the troughs** — too many people scheduled when demand is low, usually because shifts were chosen for administrative tidiness rather than demand. That is pure cost: agents paid to be available with nothing to handle. The second is **under-staffing the peaks** — too few when demand is high, which abandons service exactly when it matters most and forces reactive overtime to patch it. A schedule that over-staffs mornings and under-staffs the evening peak manages to be both expensive and poor-serving at once, which is the worst of both worlds and more common than it should be.

The levers that close the gap are staggered start times (not everyone at 9am), part-time shifts placed on the peaks, breaks scheduled into the troughs, and shift lengths varied to the shape of the day. The goal is not perfect coverage in every interval — that is impossible with whole people on whole shifts — but minimal total over- and under-staffing across the week, weighted toward protecting the peaks where service is won or lost.

A worked illustration shows the power of staggering alone. Imagine a day with a requirement that rises to a mid-morning peak, dips at lunch, and climbs to a larger early-evening peak. Roster everyone on a single 9-to-5 shift and you are heavily over-staffed at 9am, adequate at the morning peak, over-staffed through the lunch dip, and badly short from 5pm as the evening peak builds and your people go home. Now take the same headcount and stagger it: a third starting at 8, a third at 11, a third at 1 — with the late group covering the evening. Suddenly the coverage line bends toward the curve: the early start covers the morning rise, the late start covers the evening peak, and the overlap covers the middle. Not one extra person was hired; the same people were simply placed against the shape of the day rather than the convenience of a single start time. That is the single highest-return move in scheduling, and the one most often skipped.

> *The two ways a schedule fails: over-staffing the troughs (pure cost) and under-staffing the peaks (lost service). Most bad rosters manage both at once.*

5. Shrinkage in the roster

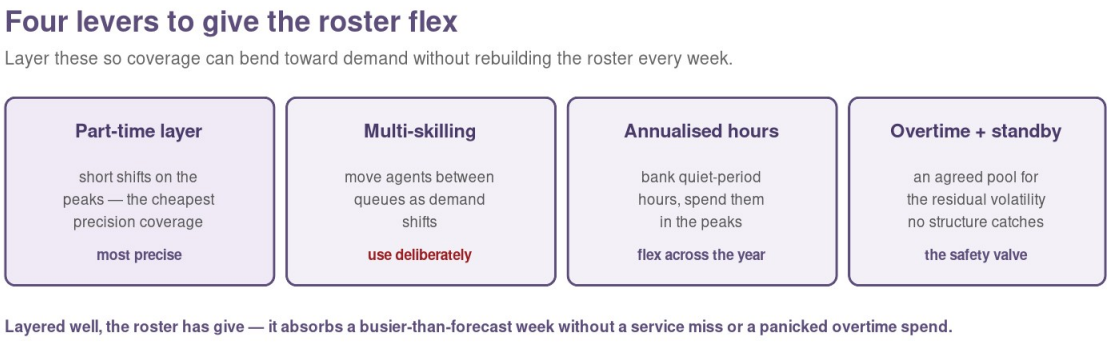
A requirement of 40 agents on the phone is not a requirement for 40 rostered agents. Between the roster and the phone sits shrinkage — holidays, training, breaks, sickness, meetings — and handling it well is what separates a schedule that holds from one that quietly falls short every day.

The gross-up is arithmetic: if 40 must be available and shrinkage runs at 30%, roughly 57 must be rostered. But two refinements matter. First, shrinkage is not flat across the day or the week — training clusters, sickness peaks on Mondays, holidays cluster seasonally — so a single annual figure applied evenly will over-cover some periods and under-cover others. Second, planned shrinkage should be placed deliberately: schedule training and meetings into the demand troughs, not the peaks, so the off-phone time lands when it costs least coverage. A schedule that lets training fall in the Monday-morning peak has turned a known, controllable activity into an avoidable service miss.

> **Design shrinkage in, don’t bolt it on**
> The common failure is to build a schedule against the on-phone requirement and add shrinkage as an afterthought, discovering only on the day that half the planned coverage is in a training room. Planned shrinkage — training, coaching, meetings, known holidays — is knowable in advance and should be placed in the roster deliberately, in the troughs, before individual assignment. Only unplanned shrinkage (sickness, no-shows) should be a same-day surprise, and even that is forecastable in aggregate.

6. The flexibility levers

No fixed roster covers a variable curve efficiently. The schedules that both serve and cost well build in flexibility deliberately, through a small set of levers that let coverage flex toward demand without rebuilding the roster each week.



The **part-time layer** is the most precise: short shifts placed exactly on the peaks, the cheapest way to add coverage where the curve is highest. **Multi-skilling** lets agents

move between queues as demand shifts, so a quiet queue's capacity serves a busy one — powerful, but with a real cost in training and in the efficiency loss of context-switching, so it pays to multi-skill deliberately rather than universally. **Annualised hours** flex capacity across the year, banking quiet-period hours for peaks. And an **agreed overtime and standby pool** handles the residual volatility that no fixed structure can. The art is layering these so the roster has give in it — able to absorb a busier-than-forecast week without either a service miss or a panicked overtime spend.

Multi-skilling deserves particular care because it is so often over-applied. The instinct is to train everyone on everything for maximum flexibility, but universal multi-skilling carries hidden costs: agents who handle many queues rarely are slower and less accurate on each, training and refresher time multiplies, and quality is harder to maintain across a wide skill set. The efficient pattern is usually a core of single-skilled specialists for the high-volume queues plus a smaller pool of multi-skilled agents who flex to wherever the demand lands — enough flexibility to absorb variation, without paying the context-switch cost across the whole workforce. The relevant question is never "should we multi-skill" but "exactly where, and how many," answered against the demand variability of each queue.

7. Designing schedules people will keep

A schedule that covers the curve perfectly and that the workforce hates is not a good schedule, because it drives the attrition and absence that destroy coverage from the other direction. The human side of scheduling is not soft; it is a coverage issue. Schedules that ignore people's lives produce sickness, no-shows, and leavers — and every leaver is a hole in the roster and a fresh recruitment cost.

Three principles make a schedule sustainable. **Predictability**: people can build a life around a roster they know in advance, and stability itself reduces attrition — late-published or constantly-changing schedules are a leading cause of avoidable leaving. **Fairness**: the unpopular shifts (evenings, weekends, splits) shared equitably rather than always landing on the same people, ideally through transparent rules or self-rostering. **Voice**: some genuine input into when people work, through preferences or self-rostering, which lifts engagement and adherence at once. A schedule designed with these in mind is adhered to better, which means the coverage on paper is closer to the coverage on the floor — and adherence-friendly design is therefore a planning lever, not an HR nicety.

> *A roster the workforce hates destroys coverage through attrition and absence. Adherence-friendly design is a coverage lever, not an HR nicety.*

There is a genuine tension here, and pretending otherwise helps no one: the schedule that fits the curve most tightly and the schedule the workforce most prefers are rarely the same roster. Perfect coverage would mean split shifts on every twin-peak day, late finishes for everyone, and start times that move with the forecast — and a workforce rostered that way would leave. The craft is finding the roster that covers well enough while remaining one people will keep, and that balance point sits a little away from pure coverage optimisation. The operations that get it right treat the trade explicitly: they quantify what a more popular roster costs in coverage, and what a tighter roster costs

in attrition, and choose deliberately rather than optimising blindly for either. A scheduler who only ever hears from the coverage report, and never from the people on the shifts, is steering with one eye shut.

8. Measuring coverage honestly

A schedule, like a forecast, is only as good as the honesty with which it is measured. The headline question is simple: how well did scheduled coverage match the requirement, interval by interval? But the way that is measured determines whether the answer is useful.

Two measures matter together. **Over-staffing** — intervals where scheduled exceeded required — is the cost side, money spent on availability that was not needed. **Under-staffing** — intervals where scheduled fell short — is the service side, where the target was at risk. Reporting only the net (they roughly cancel) hides both: a schedule that is wildly over in the morning and wildly under in the evening can look balanced on average while serving badly and costing too much. Measure the two separately, weight the under-staffing toward the peak intervals where it does most damage, and track the trend. A good scheduling function drives both numbers down over time, and watches that improving one does not quietly worsen the other.

> **Coverage is not the same as service level achieved**
> It is tempting to judge a schedule by the service level the operation actually hit. But achieved service mixes the schedule with everything that happened on the day — adherence, real-time decisions, an unexpected volume spike. To judge the schedule itself, measure planned coverage against the requirement it was built for, separately from how the day unfolded. The companion spreadsheet does exactly this: it lays scheduled agents against the requirement curve and shows the over- and under-staffing the schedule designed in, before the day even starts.

9. Common scheduling mistakes

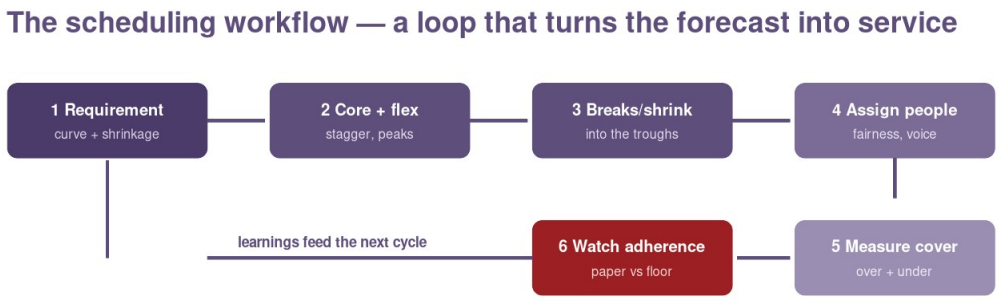
The same scheduling errors recur across operations. Knowing them is the cheapest way to avoid them.

- **Scheduling to the average, not the curve.** Flat coverage against variable demand — over in the troughs, under in the peaks.
- **Everyone starting at 9am.** Administrative tidiness that ignores the intraday shape and misses the early and late peaks.
- **Bolting shrinkage on at the end.** Building to the on-phone number and discovering the training-room hole on the day.
- **Letting breaks fall in the peak.** Lunches at noon when noon is the busiest interval — controllable coverage thrown away.

- **Over-multi-skilling.** Training everyone on everything, paying the context-switch efficiency cost everywhere for flexibility you only need in places.
- **Fitting people before fixing the shape.** Optimising for the roster nobody complains about rather than the one that covers.
- **Ignoring the human side.** Perfect coverage on paper, driven by attrition and absence into poor coverage on the floor.
- **Judging the schedule by achieved service.** Conflating the roster with everything else that happened on the day.

10. A practical scheduling workflow

Pulling the craft together, a reliable scheduling workflow runs in a repeatable loop, much as forecasting does.



Done consistently, this loop turns a good forecast into reliable, affordable service.

Start from the **requirement curve** the forecast produced, grossed up for shrinkage by period. Lay the **core coverage**, then the **flex layer**, staggering starts and placing part-time shifts on the peaks. Schedule **breaks and planned shrinkage** into the troughs. **Assign individuals** to shifts, honouring fairness and preference within the working shape. **Measure planned coverage** against the requirement — over and under, separately, weighted to the peaks — and feed what you learn into the next cycle. Then watch **adherence**, because the gap between the schedule on paper and the coverage on the floor is the final, often largest, source of error — and a signal that the schedule may need to fit the people better. Done consistently, this loop turns the forecast into reliable, affordable service.

Conclusion: build to the curve, design for the people

Scheduling is where forecasting becomes service, and it rewards the same disciplines: build in the right order, match the work to the real shape of demand, design in shrinkage and human reality rather than bolting them on, and measure honestly. The schedules

that fail are built for convenience or for the tool’s defaults; the ones that succeed are built to the curve and for the people who work them.

> *Build to the curve, design in shrinkage and the people, measure coverage honestly — and the forecast finally turns into service the operation can afford.*

The patterns and the flex levers will keep evolving, and the WFM tools will keep improving at the optimisation maths. None of it changes the craft. Lay coverage to the demand curve, build flexibility in deliberately, design schedules people will actually keep, and measure the over- and under-staffing you designed in — and you will turn good forecasts into good service, reliably, at a cost the operation can sustain.

Appendix: using the coverage spreadsheet

This paper has a free companion tool — the schedule coverage workbook, in the templates section at ccplanning.net. It lays a set of shifts against an intraday requirement curve and shows, interval by interval, where the schedule is over, under, or on target — the over/under-staffing the roster designs in before the day begins. Everything is live formulas; change a shift start time or add a part-time shift and the coverage updates.

Read the requirement curve first. The sample data gives a typical twin-peak day — a mid-morning rise, a lunchtime dip, an early-evening peak. That shape is what the shifts have to cover; a flat roster against it is the failure the paper describes.

Watch the coverage line track (or miss) the curve. The workbook sums the scheduled agents present in each interval and compares to the requirement, flagging over- and under-staffing. Try moving a shift's start time, or swapping a full-time shift for two part-timers on the peaks, and watch the under-staffing on the evening peak shrink.

The lesson in the numbers

The workbook reports total over-staffing and total under-staffing separately, plus a coverage score. The instructive exercise is to improve one without worsening the other — to shave the evening under-staffing without ballooning the morning over-staffing. That trade-off, made interval by interval, is the scheduling craft in miniature, and it is far easier to feel by dragging shifts around the model than to grasp from prose.

About ccplanning.net

ccplanning.net is an opinionated, practitioner-focused resource for contact centre workforce planning — forecasting, scheduling, real-time management, capacity planning, MI, and the leadership of the planning function. It publishes free articles, browser-based planning calculators, and a fortnightly newsletter for working planners.

This is the fifth paper in a series, and the second of the craft pair. **Paper four** covered forecasting — producing the requirement, with a worked-methods spreadsheet. **This paper** covers scheduling — meeting that requirement, with a coverage spreadsheet. Earlier papers covered AI, the business case for planning, and building a planning function. All are free at ccplanning.net.

> **Put it into practice**
>
> Download the free schedule coverage workbook in the templates section at ccplanning.net, and pair this paper with the articles on the scheduling-methods series — annualised hours, self-rostering, multi-skill scheduling, the part-time layer — and on adherence and conformance.