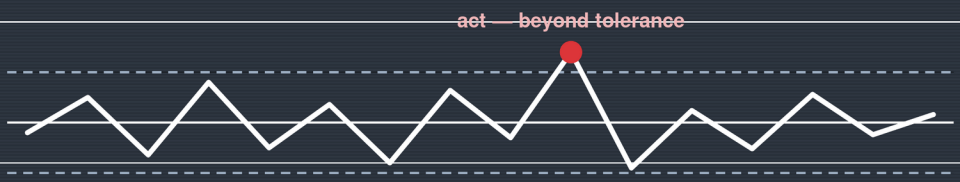
ccplanning.net

workforce planning, done well



A CCPLANNING.NET WHITE PAPER · PART

Effective Real-Time Management

Two models that both work — the dedicated team and the playbook-led approach — and how to choose.

January 2027 · ccplanning.net

Contents

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

Executive summary

Real-time management is the discipline of steering the operation through the day as it actually unfolds — closing the gap between the plan and reality as it opens. It is the most visible part of workforce planning and the most misunderstood, because the industry quietly assumes there is one right way to do it: a dedicated real-time team watching a wall of dashboards. There is not. There are at least two models that both work, and choosing the wrong one for your operation wastes either money or service.

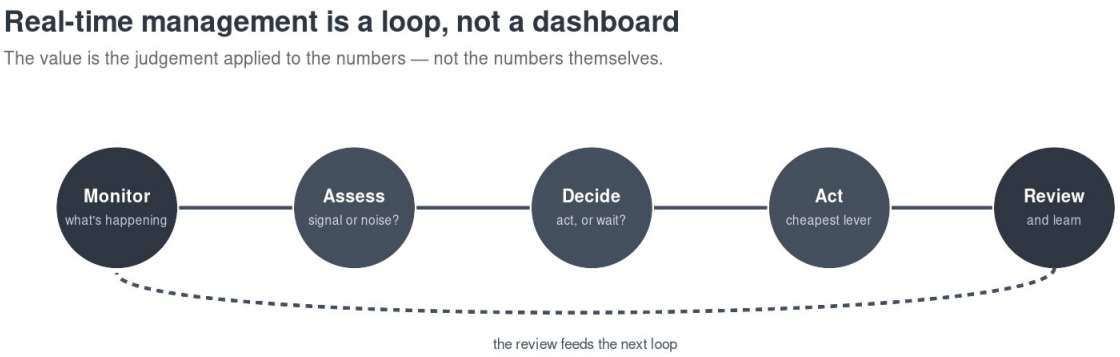
This paper sets out both. The first is the **dedicated real-time team** — a specialist function watching and steering the operation continuously. The second is the **playbook-led approach** — minimal dedicated real-time resource, with operations running pre-agreed playbooks for the situations that recur. Both can deliver excellent real-time control. The dedicated team suits large, complex, volatile operations; the playbook-led model suits smaller or more stable ones, and often outperforms a half-resourced dedicated team. The paper covers the foundation both rest on, how each works, how to choose between them, and how to run whichever you pick well.

> **The thesis in one paragraph**
> There is no single right way to do real-time management. A large dedicated real-time team and a minimal playbook-led approach are both legitimate, effective models — they suit different operations. What kills real-time performance is not choosing the "wrong" model but running an under-resourced version of the dedicated model: a half-staffed real-time desk that reacts to noise, chases every wobble, and never builds the playbooks that would let operations handle the routine itself. Pick the model that fits your size, complexity, and volatility — then resource it properly and run it with discipline.

This is the sixth ccplanning white paper. Earlier papers covered AI, the business case, building the function, forecasting, and scheduling. This one covers managing the day the plan meets reality.

1. What real-time management actually is

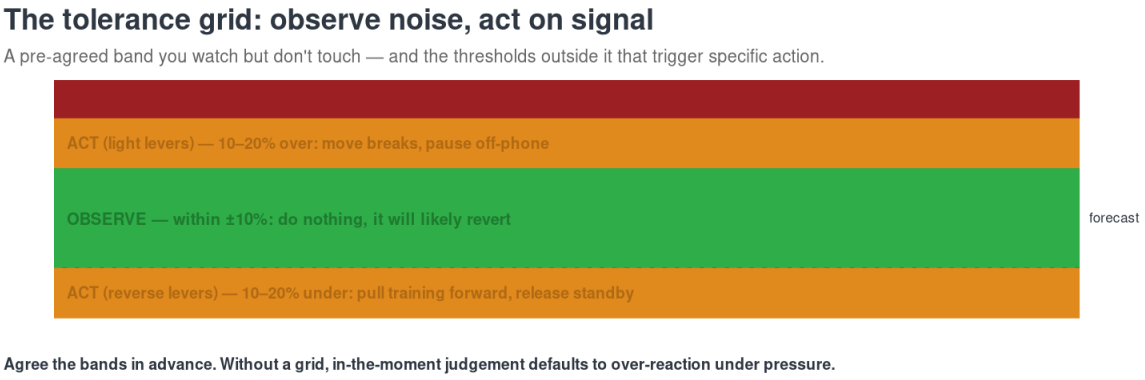
Real-time management is often reduced to "watching the dashboards," but the discipline is a loop: monitor what is happening, assess whether it is signal or noise, decide whether and how to act, act, and review what happened so the next decision is better. The dashboards are an input to that loop, not the job itself. An operation can have the best real-time dashboards in the industry and manage the day badly, because the value is in the judgement applied to the numbers, not the numbers themselves.



The purpose of the loop is to protect service and cost when the day deviates from the plan — and days always deviate. Volume arrives heavier or lighter than forecast, AHT drifts, sickness lands, a system slows, an outage upstream sends a surge. Real-time management is the operation's steering between these deviations and the plan, in the window where action can still change the outcome. The skill is knowing which deviations warrant action and which warrant patience — which is where most real-time effort is misdirected.

2. The foundation: signal versus noise

Before either model can work, the operation has to internalise the single most important truth in real-time management: most short-term variance is noise, and the most valuable real-time skill is the discipline to do nothing in response to it. This is counter-intuitive and routinely ignored, and it is where the largest real-time losses come from — not from inaction, but from over-reaction.



Consider the classic failure. A 30-minute interval comes in 14% above forecast. The analyst authorises overtime, cancels training, pulls agents off coaching. By the time those actions take effect, the variance has reverted to the mean — as most short-term variance does — but the actions persist. The operation is now over-staffed for the rest of the day, the training is rebooked, and the coaching agents are demoralised. The analyst felt productive and made things worse. The fix is a **tolerance grid**: a pre-agreed band within which variance is observed but not acted on, and outside which specific actions are triggered. The grid converts real-time from a reflex into a discipline, and it is the

foundation both models depend on. An operation that has not agreed its tolerances will over-react regardless of which model it runs.

> *The largest real-time losses come not from inaction but from over-reaction — chasing noise that reverts before the action even takes effect. The discipline to wait is the foundation both models rest on.*

3. Model A: the dedicated real-time team

The first model is the one most people picture: a dedicated real-time function — one or several analysts whose whole job is to watch and steer the operation through the day. They monitor continuously, manage the intraday re-forecast, authorise the levers, and communicate to the floor as the day develops.

Done well, this model is powerful. A skilled real-time team catches deviations early, makes fast and consistent decisions, manages complex multi-skill and multi-site operations that no playbook could fully anticipate, and frees team leaders to focus on their people rather than the numbers. The team becomes the operation's nerve centre on the day, and in a large, complex, volatile environment that nerve centre is worth far more than it costs.

But the model has real requirements, and half-meeting them is worse than not choosing it. It needs **enough resource** to cover the operating hours — a single analyst cannot watch a 24/7 operation, and a real-time desk that goes dark at 6pm is not a real-time function. It needs the **right people**: the role is roughly 40% analytical and 60% calm-under-pressure communication, and hiring only for the spreadsheet skills produces a desk that reacts badly when the day goes wrong. And it needs the **tolerance discipline** from section 2, or it becomes an expensive noise-amplifier. The dedicated team is the right answer for the operations that genuinely need it — and an expensive wrong answer for those that adopt it for appearances.

Day to day, a well-run dedicated desk has a rhythm of its own. It opens before the operation's peak with a read of the overnight position and the day's plan; it watches the intraday actuals against forecast at interval level, re-forecasting the rest of the day as the morning resolves; it holds a short check-in with team leaders at known pinch points; and it logs the decisions it makes so the post-event review has something to learn from. The good desks are quiet most of the time — watching, not acting — and decisive in the moments that clear tolerance. The busy-looking desk, forever authorising something, is usually the one managing the day worst.

When the dedicated model fits

Large operations (the team's cost is a small share of the labour base it steers); high complexity (multi-skill, multi-site, multi-channel that defies simple rules); high volatility (demand and events that genuinely need continuous human judgement); and the maturity and resource to staff it properly across the operating hours. If you cannot resource it across your hours, you do not have this model — you have a half-built version of it, which is the worst option on the menu.

4. Model B: the playbook-led approach

The second model is less discussed and, for many operations, more effective: minimal dedicated real-time resource, with the operation itself running pre-agreed playbooks for the situations that recur. Instead of an analyst deciding in the moment, the operation has already decided — in advance, calmly, with the planning team — what to do in each common scenario, and team leaders execute the playbook when the trigger is met.

The insight behind this model is that most real-time situations are not novel. The Monday-morning surge, the post-lunch dip, the system-slowdown, the unexpected-absence morning, the volume-spike-above-tolerance — these recur, and the right response to each can be worked out once, agreed, written down, and handed to the people on the floor. A playbook turns a real-time decision into a real-time execution: the team leader does not need to analyse, only to recognise the situation and follow the agreed steps. The planning team maintains the playbooks and watches the aggregate; the operation runs the day.

[missing diagram: playbook]
This model's strengths are real. It is far cheaper than a full dedicated desk. It distributes real-time capability across the operation rather than concentrating it in a desk that becomes a single point of failure. It forces the discipline of deciding responses calmly in advance rather than in the heat of the moment, which produces better decisions. And it scales down gracefully — a small operation that could never justify a dedicated analyst can run real-time well on playbooks plus a planner who checks in. Its limits are equally clear: it handles the anticipated well and the genuinely novel poorly, so it suits more stable, less complex operations, and it depends on the playbooks being maintained and the floor being trained to use them.

> *Most real-time situations are not novel. Decide the response to each recurring one once, calmly, write it down, and a team leader can execute it without an analyst in the loop.*

A worked example shows what a playbook actually contains. Take the "volume above tolerance" playbook. **Trigger:** received volume runs more than 10% above forecast for two consecutive 30-minute intervals (the tolerance grid, pre-agreed). **Step one:** confirm it is real and not a reporting glitch or a brief spike already reverting. **Step two:** move any breaks or lunches scheduled in the next hour to the cheapest later slot, returning coverage now at no cost. **Step three:** pause non-critical off-phone activity — coaching and internal meetings — and return those agents to the queue. **Step four:** if still short

after fifteen minutes, flex multi-skilled agents in from the quieter queues. **Step five:** only then request voluntary overtime. **Communicate** at each step in plain language, and **log** what was done. A team leader handed that playbook does not need to be a real-time analyst; they need to recognise the trigger and follow the steps — and the steps embody exactly the proportionate, cheapest-lever-first discipline a good analyst would apply, decided calmly in advance rather than under pressure.

5. Choosing between the two models

The choice is not about which model is better in the abstract — both are legitimate — but which fits your operation. Four factors decide it, and they usually point the same way.

Choosing the model: four factors

They usually point the same way. The model follows the operation, not fashion.

Factor	Points to dedicated team	Points to playbook-led
Size	large — team cost is a small share	small — a desk is too big an overhead
Complexity	multi-skill / site / channel	single-skill, single-site
Volatility	high, genuinely unpredictable	stable, predictable
Resource / maturity	can staff full hours + tolerance discipline	cannot resource a desk properly

Size. A dedicated team is a fixed cost; it only makes sense when it is a small share of the labour base it steers. A 40-seat operation cannot justify two real-time analysts; a 2,000-seat one easily can. **Complexity.** Multi-skill, multi-site, multi-channel operations generate situations no playbook fully anticipates, favouring human judgement; a single-skill, single-site operation is far more playbook-able. **Volatility.** Operations with high, genuinely unpredictable variation need continuous judgement; stable, predictable ones can pre-decide most responses. **Maturity and resource.** A dedicated team only works if it can be staffed across the operating hours and run with tolerance discipline; without that, the playbook model will outperform it.

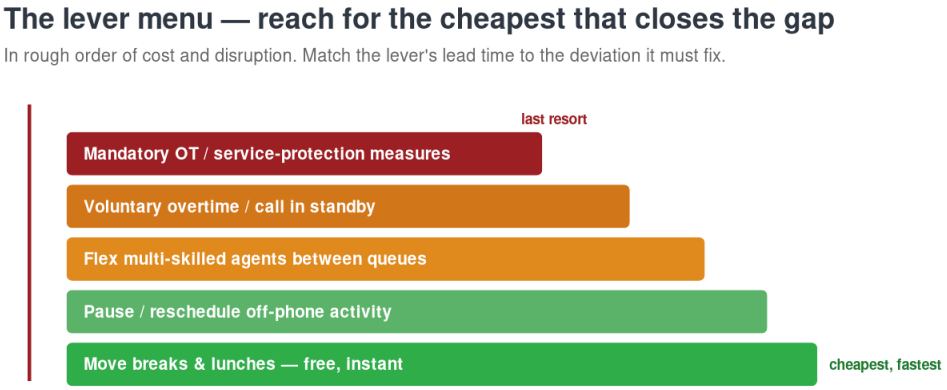
Most operations are not at the extremes, and the best answer is often a **blend**: a small dedicated real-time presence for the complex, judgement-heavy decisions, plus playbooks that let the floor handle the recurring situations without escalating. The blend captures the cheapness and distribution of playbooks for the routine and reserves scarce analyst attention for the genuinely novel. The mistake is to default to a dedicated team because that is what real-time "looks like," resource it for appearances, and end up with the worst of both — an expensive desk that still chases noise because it never built the playbooks.

A worked choice makes it concrete. A 120-seat, single-site, mostly single-skill operation with stable weekday demand asks whether it needs a dedicated real-time analyst. Run the factors: size is modest (one analyst would be a meaningful share of the overhead), complexity is low (single-skill, single-site), volatility is moderate and largely predictable,

and the operation is small enough that team leaders know the floor intimately. The factors point clearly to the playbook-led model — a set of well-built playbooks for the half-dozen recurring situations, maintained by the existing planner, with team leaders executing them. A dedicated analyst here would spend most of the day watching a stable operation and, lacking enough to do, would tend to over-manage it. Now change one variable — make it 1,200 seats across three sites with five skills and campaign-driven volume — and the same analysis flips: the complexity and volatility now genuinely need continuous judgement, and the analyst cost is trivial against the labour base. Same question, opposite answer, because the operation is different. That is the whole point: the model follows the operation, not fashion.

6. The lever menu: what real-time can actually do

Whichever model runs it, real-time management acts through a finite menu of levers, and knowing the menu — and the cost and lead time of each — is what turns a deviation into a sensible response rather than a panic.



The levers fall into rough order of cost and disruption. **Cheapest first:** adjust break and lunch timing to move coverage within the day at no extra cost; pause or reschedule off-phone activity (training, coaching, meetings) to return people to the phones temporarily. **Then:** flex multi-skilled agents between queues to move capacity to where demand landed; adjust the channel blend if agents handle more than one. **More disruptive:** request voluntary overtime or call in standby; extend or shorten shifts by agreement. **Last resort:** mandatory overtime, or service-protection measures like adjusting the IVR or messaging. The discipline is to reach for the cheapest lever that will close the gap, in proportion to a deviation that has cleared the tolerance band — not to leap to overtime for a wobble that will revert before the overtime starts.

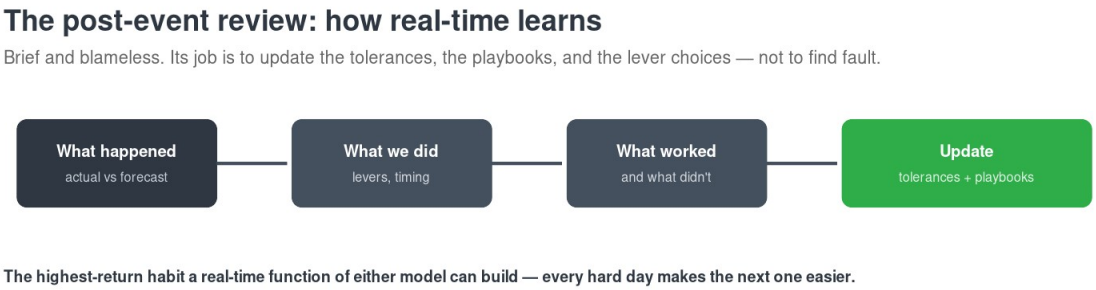
Two things about the menu are easy to miss and worth stating. First, each lever has a **lead time** as well as a cost: moving a break takes effect in minutes, voluntary overtime in tens of minutes to hours, and a mandatory call-in longer still — so a lever is only useful if its lead time is shorter than the deviation it is meant to fix. Authorising overtime to cover a one-hour spike that will be gone before anyone arrives is pure waste, a mistake the lead-time lens prevents. Second, the menu runs in both directions: when a day comes in **under** forecast, the same levers reverse — bring breaks and training forward into the unexpected quiet, release standby, offer early finishes — turning an over-staffed day into

banked development time or saved cost rather than idle agents. Real-time is as much about spending an unexpected lull well as about covering an unexpected peak.

7. Communication and the post-event review

Two practices separate real-time functions that improve from those that simply react, and both apply whichever model you run.

The first is **communication on the day**. When the day is going wrong, the operation looks to whoever owns real-time for calm, clear direction — and how that is delivered matters as much as the decision itself. A real-time analyst or a team leader running a playbook should tell the floor what is happening, what is being done, and what is expected of them, in plain language and without panic. The 60% of the real-time role that is communication rather than analysis lives here: a technically correct decision delivered as alarm produces a worse outcome than a good-enough decision delivered with calm authority.



The highest-return habit a real-time function of either model can build — every hard day makes the next one easier.

The second is the **post-event review**. After a significant day — a major event, a bad miss, an unusual surge — the operation should look back: what happened, what was forecast, what real-time did, what worked, and what the playbooks or tolerances should learn. This is the loop closing on itself. Without it, real-time repeats the same over-reactions and the playbooks never improve; with it, every difficult day makes the next one easier to handle. The review is brief and blameless — its purpose is to update the tolerances, the playbooks, and the lever choices, not to find fault. It is the single highest-return habit a real-time function of either model can build.

8. Common real-time mistakes

The same real-time errors recur across operations and across both models.

- **Reacting to noise.** Acting on every deviation instead of waiting for variance to clear the tolerance band — the costliest and most common error.
- **The half-built dedicated desk.** Choosing the dedicated model for appearances, then under-resourcing it so it cannot cover the hours or build the playbooks.
- **No tolerance grid.** Leaving "when do we act" to in-the-moment judgement, which defaults to over-reaction under pressure.

- **Leaping to expensive levers.** Authorising overtime for a wobble when a break-timing adjustment would have closed the gap.
- **Hiring only for the spreadsheet.** Staffing a real-time desk for analytical skill and neglecting the calm-communication 60% of the job.
- **No playbooks in a playbook-able operation.** Forcing analyst judgement on situations that recur and could have been pre-decided.
- **No post-event review.** Repeating the same mistakes because the difficult days are never learned from.
- **Alarm in communication.** Delivering real-time direction as panic, which spreads to the floor and worsens the day.

Conclusion: pick the model, then run it with discipline

Effective real-time management is not about owning the most impressive real-time desk. It is about choosing the model that fits your operation — the dedicated team for the large, complex, volatile environments that need it; the playbook-led approach for the smaller or more stable ones, and often a blend — and then running it with the discipline that makes either work: a tolerance grid that filters noise, a lever menu used in proportion, calm communication, and a post-event review that closes the loop.

> *Both models work. What fails is the under-resourced middle — a half-built dedicated desk that chases noise and never builds the playbooks. Choose deliberately, resource properly, run with discipline.*

The dashboards will keep improving and AI will keep getting better at flagging the deviations worth attention. None of it changes the core: real-time management is judgement applied to deviation, in proportion, with calm. Decide which model fits, give it the tolerance discipline and the lever menu and the review habit, and the operation will steer through its difficult days well — whichever way you choose to run the desk.

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 sixth paper in the series. Earlier papers covered AI in planning, the business case for planning, building a planning function, the forecasting craft, and the scheduling craft — most with a companion calculator, assessment, or spreadsheet. All are free at ccplanning.net.

> **Read next**
> Pair this paper with the articles on the discipline of doing nothing in real-time, real-time playbooks, hiring a real-time manager, and the real-time lever menu — all free at ccplanning.net.