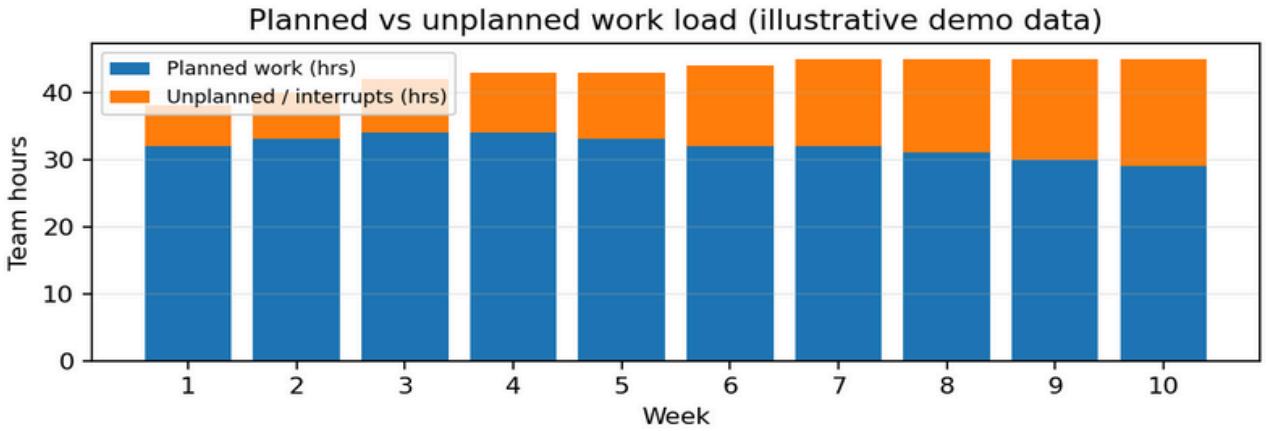
Interruptions Kill Predictability

The hidden delivery tax of urgents, priority churn, and context switching (3 pages).

Page 1 — The finding

Most delivery misses aren't caused by bad estimates. They're caused by **unbounded interruptions**: urgent asks, mid-sprint reprioritization, meetings, and "quick" side requests that fragment focus and create queues.

> **Rule of thumb:** If unplanned work grows for 2+ weeks while planned capacity is flat, the delivery date is already moving—even if status stays green.



When the unplanned band grows, the schedule is governed by interrupts—not your plan.

The three signals to track weekly

Signal	How to spot it	Why it predicts drift
Unplanned work %	More interrupts/escalations than last week	Reduces effective capacity + increases tail risk
Priority churn	Top priorities change mid-week (often)	Creates rework and context switching
Aging in-progress work	Items sit "in progress" without movement	Queues form; queues control ship dates

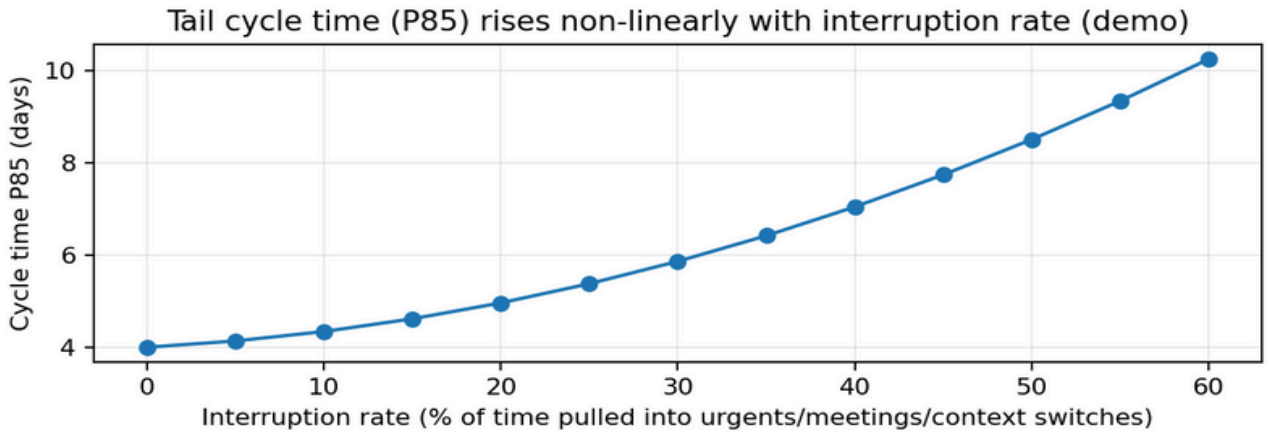
Page 2 — Why it happens

Interruptions don't just consume time. They create a compounding effect: context switching overhead, partially finished work that blocks others, and longer feedback loops. The result is a fat tail—**a few items take much longer than expected**, and those items are what break dates.

> **Rule of thumb:** Treat interruption rate like a budgeted expense. If it's unbounded, delivery is unbounded.

A simple model leaders can use

Effective delivery capacity ≈ Planned capacity × (1 – interruption rate). But the real damage is in the tail: as interruption rate rises, cycle time percentiles (P85/P95) rise faster than linearly.



You can feel "only a bit busier," while tail cycle time doubles—and the ship date is controlled by the tail.

10-minute diagnostic

Question	What to look for
Is unplanned work rising?	Unplanned hours or tickets trending up 2+ weeks
Are priorities stable?	"New top priority" more than once/week
Do items pause and wait?	Oldest in-progress items aging without movement
Is the day fragmented?	Meetings split the day into small blocks; little uninterrupted time

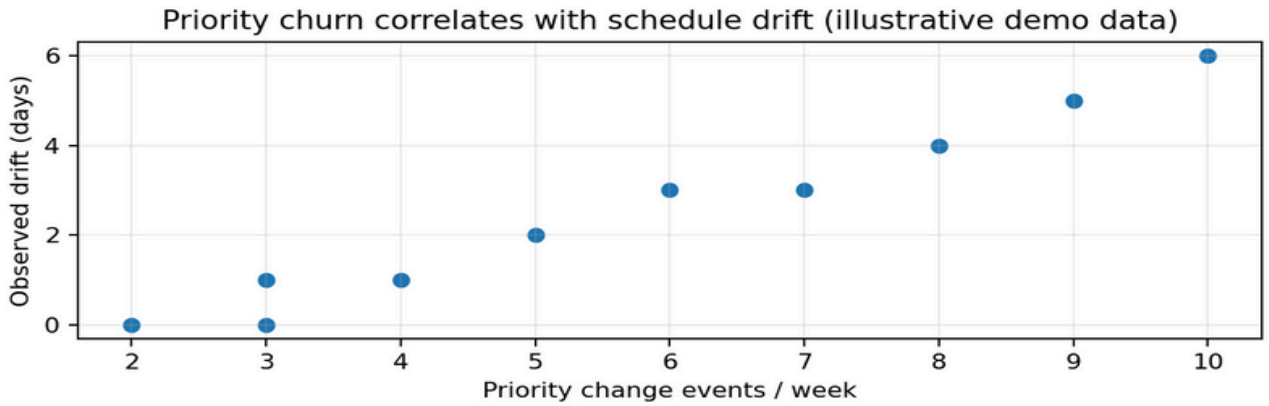
Page 3 — Fixes that actually reduce drift

The goal is not “fewer emergencies”. The goal is a system where emergencies are absorbed without destroying the plan. That requires explicit policies: an **interruption budget**, a **triage gate**, and **protected focus windows**.

The interruption budget playbook

Lever	What to do	Expected impact
Set an interrupt budget	Cap unplanned work (e.g., 10–20%). Track weekly.	Stabilizes capacity assumptions
Triage gate	One rotating owner filters urgents; not everyone context-switches.	Cuts thrash + duplicate work
Freeze window	Define periods where priorities can’t change except Sev-1.	Reduces rework + protects deep work
Finish-first rule	If budget is exceeded: stop starting, swarm to finish aging work.	Shrinks queues + tail risk

Why churn correlates with drift



Churn is not “agility.” Past a point, it’s schedule debt being created in real time.

Motionode’s solution

Teams can assemble these signals manually, but it usually takes weeks and goes stale fast. Motionode computes interruption signals (unplanned work, churn, aging) and lets leaders run what-if scenarios to recover the date with the least cost.