CASE STUDY · DELIVERY PLATFORM · UENGAGE

Owning Flash

A delivery platform with four audiences, and the job of deciding which failures it absorbs quietly

Product Manager at uEngage, owning Flash. Dec 2024 to Present 8 min read

---

backpocket.website/studies

IN SHORT

- Own Flash, the delivery platform at uEngage: own fleet management, third-party logistics integrations, the rider app, the support systems and the interfaces merchants and the people ordering both touch.
- Defined the own fleet strategy end to end: the North Star metric, the staffing model and the phased rollout for a mixed full time and gig fleet.
- Introduced On-Time Delivery Rate as the metric the team watches, moving it from under half of orders to a little over three quarters.
- Cut daily escalations by roughly 60% while throughput more than tripled, and brought escalation resolution down from five working days to 24 hours.

Four parties, one system

A delivery platform looks like one product from the outside. Four parties sit on it: the person waiting for dinner, the merchant who has to make it, the rider carrying it, and the operations team who hears about it first when any of the three breaks. Their incentives do not agree.

I own Flash, the delivery platform at uEngage. That covers own fleet management, third-party logistics integrations, the rider app, the support systems and the interfaces the merchant and the person ordering both touch. One system carries all four.

Four audiences on four products is a portfolio problem. Each product picks its primary audience, keeps its own backlog and gets its own definition of better, and the trade between them is made a level above the products.

Four audiences on one system is an allocation problem. The trade sits inside the same release: a change made for the merchant is charged to the rider, and a screen that keeps the operations team out of a call is one more screen

the merchant has to read. There is no neutral default.

The four do not report failure at equal volume either. An operations team escalates, a merchant calls, and somebody waiting for dinner is more likely to leave than to write in. Volume of complaint is not the same as cost of failure, and a backlog built from the loudest party will get that wrong.

There is no staging environment for a hungry customer

Most software failures are recoverable, and most of them are quiet. A delivery failure is physical and time-boxed.

A mis-assigned rider is somebody's birthday cake at the wrong address. A regression at 8pm is a dinner that never arrives.

A software bug waits for its fix. It sits there, it reproduces, and the fix retires it for everybody who would have hit it next. A delivery failure expires before it can be diagnosed, and the fix lands on the next order rather than the one that broke.

That splits the work in two. One track is the fix, which behaves like software and can be planned and queued. The other is what the system does inside the hour, with nobody in the loop, while the order is still live.

It also breaks the severity ladder that software brings with it. Severity in software counts how many are affected. A late order affects one person, completely, and counting by reach puts the whole of somebody's evening in the lowest band.

Decide what the system absorbs, and what it has to say out loud

Most of the job is deciding which failures the system should absorb quietly and which it has to surface loudly. Absorbing a failure somebody needed to know about breaks trust; surfacing every wobble turns the operations team into a switchboard. Both directions cost something.

The test I use is not how bad the failure is. It is whether the promise that was made can still be kept. A failure the platform can still deliver against is the platform's to handle. A failure that changes what somebody was told belongs to that person, straight away.

Attention on an operations floor is fixed for the day. Every event surfaced spends some of it, so a system that surfaces everything has pushed the decision onto whoever is on shift.

What came out of that framing:

- Automated escalation handling, so a known failure moves without waiting for somebody to notice it.
- Pushed proactive updates out to the person waiting before the wait turns into a call.
- Shipped live tracking for orders in flight.
- Established single channel triage with clear ownership for production escalations.

Grow the integration surface, and take the manual step out of onboarding

Third-party logistics integrations went from four to sixteen. Sixteen rather than four is more places an order can go, and it is twelve more sets of failure behaviour to normalise into one flow.

Counted as an integrations project that is twelve pieces of delivery work. It is a supply decision. An order nobody accepts is not a slow order, it is an order with nothing behind it, and no amount of tracking or messaging will produce a rider.

The integration is the price rather than the point. Each partner reports a pickup differently, cancels differently and goes quiet differently. All of it has to arrive in one shape at the other end: nobody on the merchant side is going to learn sixteen vocabularies.

POS integrations across partner platforms carry order flow automatically, and manual merchant onboarding is out of the process. A manual onboarding waits on somebody being free to do it. It also caps the number of merchants who can come on in a day at whatever that person can get through.

Run the fleet where the failure cannot be absorbed

The own fleet came out of the same question. An order with no supply behind it cannot fail quietly: there is nobody to reassign it to and nothing to route it through, so the person waiting is the one who finds out.

I defined the North Star metric for it, the staffing model and the phased rollout. The fleet runs as a mixed model, full time and gig.

Demand for dinner is not flat across a day. A fleet sized for the peak stands still through the afternoon, and a fleet sized for the average is short exactly when it matters. A mixed model answers that shape.

Rider assignment used to sit in the low seventies as a percentage, and slipped under that on bad days. Assignment is the first place where absorbed failure either exists or does not, because an unassigned order has no quiet path left. Assignment holds in the mid nineties now.

Watch the number that lands on a person, not the average

I introduced On-Time Delivery Rate as the metric the team watches. An average delivery time is a number nobody experiences: it is made of orders that arrived early and orders that arrived late, and it hides both. A rate lands on a person rather than on a chart.

It has gone from under half of orders to a little over three quarters. The remaining quarter is orders where somebody was promised one thing and got another.

What the rate rules in is the tail. Every order scores once against the promise it was given, so a bad hour shows up as the orders it spoiled.

What it rules out is severity. Two minutes late and an hour late score as the same failure, and the rate will not separate a slow kitchen from a slow road. It is also only as honest as the promise underneath it, because a longer promise raises the rate without a single order arriving sooner.

Take escalations down without taking orders down

Daily escalations came down by roughly 60% while throughput more than tripled. The second figure is why the first one counts, because a support queue also empties when a business shrinks. This is the one I care about most.

The two move in opposite directions, which is the part that is difficult to manufacture. The count fell against a base that grew, so escalations per order fell by more than the count did.

It does not prove the failures stopped. An escalation is a failure somebody reported, and automated handling moves a known failure before anybody has to report it. Part of that fall is fewer failures and part of it is fewer failures reaching a person.

Escalation resolution went from five working days to 24 hours. Single channel triage sits under that: every production escalation goes to one place with one owner, so the time is spent on the fix rather than on finding who owns it.

My reading is that most of what that removed was not repair work but the hours before the repair started. That is a reading rather than a measurement.

Read each movement against what it rules out

Each measure here was chosen for what it refuses to flatter as much as for what it shows.

MEASURE	MOVEMENT	WHY THIS WAS THE MEASURE WORTH MOVING
On-Time Delivery Rate	From under half of orders to a little over three quarters	An average hides the early and the late together. A rate scores every order once against the promise it was given, so the tail is the thing being watched.
Rider assignment	From the low seventies to the mid nineties, as a percentage	Assignment is the gate. Nothing downstream can absorb a failure on an order that has not been picked up, so an unassigned order is the one case with no quiet path.
Daily escalations	Down by roughly 60%, against throughput more than tripling	A falling queue on its own is also what a shrinking business looks like. The pair is the claim rather than the fall.
Escalation resolution	From five working days to 24 hours	Most of that time was going on routing rather than on repair. One place and one owner is the shortest thing that removes it.
Third-party logistics integrations	From four to sixteen	Counted as integrations it is delivery work. It is supply: an order nobody accepts cannot be fixed by tracking or by messaging.

Name what the numbers cannot show

Every figure here is a movement rather than an absolute, and that is a choice. Order volumes, merchant counts and unit economics belong to the company that paid for them, so those are not on this page.

Nothing on this page counts the failures the system absorbed. An absorbed failure leaves no escalation to count and no late order to score, so that class of outcome has no number beside it.

The cost of that lands on whoever is reading this. A band carries the direction and the rough size, and it does not carry the base underneath, so no figure here can be checked against a market.

What I take from it

- Four audiences on one system is an allocation problem rather than a prioritisation problem, because every change is charged to somebody.
- Most of the job is deciding which failures the system absorbs quietly and which it has to surface loudly.
- A fall in escalations means nothing on its own, because a support queue also empties when a business shrinks.
- Movements can be published where absolutes cannot, and the reason for that belongs on the page.

> Every business figure on this page is a movement rather than an absolute. Order volumes, merchant counts, revenue and service levels belong to the company that paid for the work, so those are not here. Counts that describe what was built rather than how the business performs, like the number of logistics partners integrated, are mine to state.