

CASE STUDY

Service Management That Outlives the Tooling

Service mapping, a CMDB that sticks, and incident response built to survive the Opsgenie sunset.

Client Multi-region logistics & fulfilment operator

Industry Logistics & supply chain

Engagement JSM foundation • service mapping & CMDB • incident and problem management

Window 16 weeks — major incident management live at week 9



Solution Partner

The Situation

The client ran fulfilment around the clock across multiple regions — and ran it on a Jira Service Management instance that had quietly become a ticket queue. Incidents got worked. Nobody could say what they touched.

They asked Avaratak to:

- 1 Build a service model the business would recognise — and a CMDB behind it that survives its own launch
- 2 Mature incident and problem management to an ITIL-aligned standard, major incidents included
- 3 Put alerting, on-call and customer status communications on a footing that lasts

Stakeholders: VP of Infrastructure, Director of Service Delivery, application owners across WMS and TMS, and the Customer Experience lead.

What leadership could feel

Every major incident opened with a scavenger hunt

Who owns this? What depends on it? Who do we have to tell?

Two CMDB attempts had already died as spreadsheets

Both were inventories. Neither answered a question anyone asked at 2am.

Customers found out from their own customers

Status went out late, by hand, and only when someone remembered.

The Avaratak Difference

APRIL 5, 2027

Atlassian ends support for Opsgenie. New sales stopped in June 2025.

The brief asked for an Opsgenie build-out. We showed them the sunset date instead — new schedules and escalation policies there would only get built twice. So we built alerting and on-call natively in Jira Service Management: same capability, one migration avoided, and a smaller invoice than the one they arrived ready to sign.

Senior Atlassian expertise, bespoke solutions, boutique responsiveness and cost transparency — small team, big results, no surprises.

Service-first, never inventory-first

Most CMDB programmes die trying to catalogue everything. We modelled the services the business names during an outage, then hung configuration items off them. Scope you can finish beats scope you can admire.

The consultants who scoped it, built it

The people who ran the mapping workshops wrote the schemas. No discovery-to-delivery handoff, no junior relearning your estate on your budget.

Every number carries its method

Each result we report names how it was measured and when it gets measured again. If we can't tell you how we counted it, it doesn't go on a slide.

Our Objectives

01

Make the service model real

A named, owned catalogue of business and technical services — not a wiki page that goes stale in a quarter.

02

Build a CMDB that survives launch

Configuration items tied to services, with governance that keeps the data honest long after we hand over the keys.

03

Mature incident and problem

Major and minor incident paths, escalation and comms that run without heroics, and root cause analysis that closes the loop.

04

Future-proof the operational stack

Alerting, on-call and status communications on products that will still be supported three years from now.

Our Approach: The Service Model & the CMDB

Atlassian's own guidance is to start a CMDB from services, not from servers. We took it literally.

1 Service discovery workshops

Six facilitated sessions with business and technical stakeholders in the same room. Output: business services the company sells, technical services IT runs, and the applications underneath both.

2 Ownership before attributes

Every service got a named owner and an escalation path before we modelled a single attribute. A configuration item with no owner is a rumour with a hostname.

3 Assets schema design

Object schemas, object types and the relationship model built in JSM Assets: service → application → infrastructure → vendor, with dependencies mapped in both directions.

4 Governance built in, not bolted on

CI lifecycle states, a quarterly ownership review, a change gate that keeps records true, and a data quality dashboard the service owners actually see.

Our Approach: Incident, Problem & Readiness

With the service model in place, severity stopped being a debate and started being a lookup.

1 Major and minor incident management

A severity matrix keyed to service criticality from the model, MIM roles, a bridge process, and a communications cadence that runs on a timer instead of on somebody's nerve.

2 Problem management that closes the loop

Problem records linked to their incidents, structured root cause analysis templates, a known error register, and a monthly review that reads the RCAs rather than filing them.

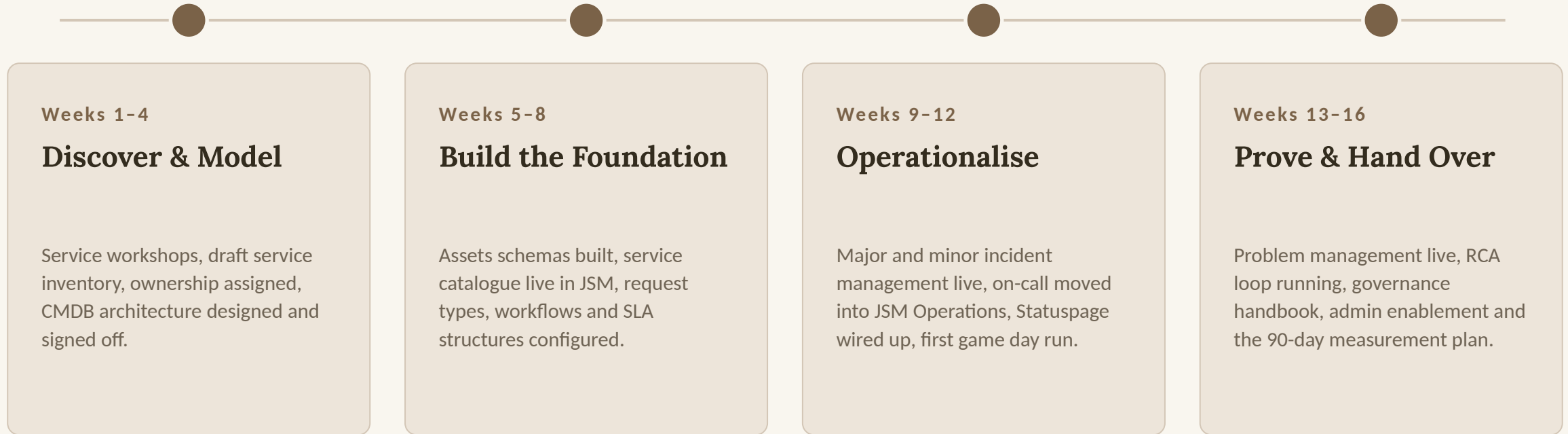
3 Alerting and on-call in JSM Operations

Schedules, rotations, escalation policies and monitoring integrations built natively — deliberately not in Opsgenie, and mapped to the same services as the CMDB.

4 Status communication, triggered not typed

Statuspage components mapped one-to-one to catalogued services, with external updates and internal notifications fired from the incident record itself.

Timeline: Sixteen Weeks, Four Phases



Major incident management went live at week 9 — deliberately, and before the build was finished. The client had a peak season to survive, and a half-built capability they can run beats a complete one they get in January.

Early Outcomes

63% faster

Impact scoping on major incidents

Measured: incident raised to declared impact scope. Median fell from 38 minutes at baseline to 14 minutes at the 90-day re-measure. JSM native reporting.

94% of P1/P2

Incidents raised against a named service

Measured: service-field completion on incident records, native JSM export. Up from a 22% baseline on the legacy free-text field.

100%

Catalogued services with a named owner and on-call path

Measured: ownership coverage across the catalogue, exported from Assets. A delivery gate, not a happy accident.

7 minutes

Median time from incident declared to customer status update

Measured: Statuspage publish timestamp against incident declaration. Down from 41 minutes across the prior quarter.

What We Delivered

✓ **Service Catalogue & Portfolio**

Business and technical services, named owners, criticality tiers

✓ **CMDB Governance Pack**

Lifecycle states, review cadence, data quality dashboard

✓ **Incident & Problem Workflows**

Triage, SLAs, escalation, RCA templates, known error register

✓ **Statuspage Configuration**

Components mapped to services, publishing automated from incidents

✓ **CMDB Architecture in Assets**

Object schemas, CI types, and a two-way relationship model

✓ **Major Incident Playbook**

Severity matrix, roles, bridge process and comms runbook

✓ **Alerting & On-Call in JSM**

Schedules, rotations, escalation policies, monitoring integrations

✓ **Admin Runbook & Enablement**

Governance handbook, training, and a 90-day measurement plan

Capability Coverage

The capability areas a service management brief asks for — and what this engagement built against each.

✓ Jira Service Management

Service design and build-out, request types, queues and portal experience — plus a service catalogue with named owners and criticality tiers.

✓ Service mapping & modelling

Facilitated workshops with business and technical stakeholders; services, applications and dependencies documented, and ownership assigned.

✓ CMDB design & governance

Object schemas, CI structure and two-way relationships in Assets, aligned to the service model — with lifecycle states and a review cadence.

✓ Incident management

Severity matrix, MIM roles and bridge process; minor incident triage, SLAs, escalation and comms — tested in a game day before go-live.

✓ Problem management

Problem records linked to their incidents, structured RCA templates, a known error register, and a monthly review that closes the loop.

✓ Alerting, on-call & status

Schedules, rotations and escalation policies built natively in JSM Operations; Statuspage published from the incident record.

All of it inside one sixteen-week engagement, delivered by the same senior consultants who scoped it.

Client Voice

“

We'd tried to build a CMDB twice. Both times it turned into a spreadsheet nobody opened. The difference this time was starting from the services we argue about at two in the morning, instead of from a list of servers.

— Director of Service Delivery

“

They talked us out of work they could have billed for. We were ready to buy an Opsgenie build-out; they showed us the end-of-support date and built it in JSM instead. That is the entire reason they are still our partner.

— VP of Infrastructure

Quotations used with permission. Speaker roles retained; names, employer and region withheld at the client's request.

Key Takeaways

Service management maturity is not a tool configuration. It is a set of agreements about what you run, who owns it, and who answers for it when it breaks. The tooling only makes those agreements enforceable.



- 1 Start where the argument starts**
Modelling the services the business names during an outage gave the CMDB a finish line — and a reason for anyone to keep it accurate.
- 2 Don't build on a published end date**
The cheapest migration is the one you never run. Checking a product roadmap before configuring it is not clever; it is just the job.
- 3 The client can now answer three questions in under a minute**
What is broken, what does it touch, and who is fixing it. That used to take an hour and a conference bridge.

Ready to Find Out What You Actually Run?

About Avaratak

Avaratak is an Atlassian Solution Partner delivering enterprise-grade expertise with personalised service and honest pricing — from licensing through implementation, training and ongoing improvement.

We bring senior Atlassian consultants, transparent pricing, and a stubborn commitment to building things that still work after we leave.



Solution Partner

Start With a Service & CMDB Readiness Review

Two weeks. We map your top services, test whether your configuration data can answer a real incident question, and give you a written view of where the gaps are.

You get the findings whether or not you engage us for the build. If your CMDB is healthier than you feared, we will happily tell you that and hand back the rest of the budget.

Book a Conversation

avaratak.com