

PROCESS SAFETY REPORTING

Client Case Study

Process safety reporting across generation and LPG operations

Prepared by Succinct Software Pty Ltd, Brisbane

Client One of Australia's largest energy suppliers

Engagement Process safety reporting platform, 2022 to 2026

Date September 2026

Status V1.0, public release





CONTENTS

1. Executive Summary.....	3
2. The Problem.....	4
3. How the Work Was Structured.....	5
4. What Was Built.....	6
5. Why It Held Together.....	6
6. Outcomes.....	7
7. What This Means for Other Operators.....	8
8. About Succinct Software.....	8

1. Executive Summary

Between 2022 and 2026, Succinct Software designed, built and delivered a reporting and analytics platform that gives the client a single consolidated view of process safety performance across its generation and LPG operations.

Before the platform existed, that view had to be assembled by hand. Indicator data sat across several systems and spreadsheets, and producing a monthly picture of how risk controls were performing took real effort from people whose time was better spent elsewhere. By the time a pack was finished, it described a month that had already gone.

The platform replaced that with a purpose-built application. It draws indicator data from the client's source systems, applies the client's own KPI definitions, and presents the result through dashboards built for the people who use them: operations leaders, process safety specialists, and the site teams responsible for the controls being measured.

The final component went live in August 2026. Over four years the platform grew from a single dashboard concept into a governed reporting capability spanning multiple asset classes, with Power BI integration into the client's wider reporting estate and a deployment model the client's own IT function runs.

This case study sets out what the problem was, how the work was structured, what was built, and what we would carry into a similar engagement elsewhere.

Engagement at a glance

Client	One of Australia's largest energy suppliers
Delivery partner	Succinct Software Pty Ltd
Sector	Power generation and LPG operations
Engagement period	2022 to 2026
Funding model	Capital investment, delivered through successive statements of work
Asset classes	Power generation, LPG
Core technology	React front end, containerised deployment, Power BI integration
Role	Solution architect and development manager
Delivery team	Project manager, front-end developer, back-end developer, test and QA specialist
Status	Build completed August 2026; support to November 2026

2. The Problem

Power generation and LPG handling are high-hazard operations. The organisations that run them are judged, internally and externally, on how well they understand whether their risk controls are actually working. That understanding comes from process safety indicators: the leading measures that show a control degrading before it fails, and the lagging measures that record what happened when one did.

The client had the indicators. What it did not have was a low-effort way to see them together.

The data sat across separate systems, each with its own structure and refresh cycle. Producing a consolidated view meant exporting, reconciling and reformatting by hand, then rebuilding the same pack the following month. The work was repetitive, the output was static, and the definitions behind each number depended on whoever had assembled it.

That creates two problems. The obvious one is cost: senior technical people spending days a month on data handling. The second one matters more. A manual reporting cycle is a slow reporting cycle, and a leading indicator that surfaces six weeks after the fact has quietly become a lagging one.

The goal was a single application, owned internally, that produced the same view every time, from the source data, with nobody in the middle.

That was the brief the client's data and engineering information team brought to Succinct Software in 2022.

3. How the Work Was Structured

Succinct was engaged as a delivery partner embedded in the client's own team rather than as a vendor working to a fixed specification handed over at the start. That distinction shaped everything that followed.

The platform was funded as a capital investment and delivered through successive statements of work, each scoping a defined increment. That gave both sides something worth having. The client could fund a piece of work, see it land, and decide what came next having actually used the previous release. Succinct could build against real feedback rather than against a document written before anyone had touched the product.

The work ran in six phases:

1. Functional review of the existing dashboard
2. Requirements gathering with operations and process safety specialists
3. Design and build-out, using coding templates on a common architecture pattern, with a database uplift underneath
4. Implementation into the client's environment
5. Testing, with dedicated QA across every release
6. Support, including hypercare after go-live

Build-out, implementation and testing repeated across successive releases rather than running once.

Benjamin Snelling held two roles across the engagement: solution architect, owning the technical design and the decisions that came with it, and development manager, running the people who built it. Both, for four years, on the same platform.

Around that sat a small, stable delivery team: a project manager, a front-end developer, a back-end developer, and a dedicated test and QA specialist. Keeping it small was deliberate. On a platform where the domain knowledge is the hard part, continuity of people is worth more than raw capacity, and every handover costs more than it looks like it will. The same group carried the work from the first release through to the final component four years later.

Two working arrangements that mattered

The first: the client held technical authority. Architecture decisions were reviewed and challenged by the client's own program management rather than delegated to the supplier. When the deployment approach moved from Fargate to Docker containers, it moved because the client's IT function needed a model it could operate itself. The decision came out of that conversation, not out of a supplier preference.

The second: the product carries the client's branding and runs inside the client's environment. It presents as an internal client system, because that is what it is. Nothing about it reads as a third-party attachment, and nothing about it depends on Succinct to keep running.

4. What Was Built

The platform is a web application sitting on an integration layer that draws from the client's source systems on a scheduled basis.

The front end is built in React. Its central feature is the Asset and Hazard Dashboard, which gives a consolidated view of safety-relevant indicators against specific assets: what is being measured, how it is tracking, and where attention is needed. Users move between an enterprise view and an individual asset without changing tools or exporting anything.

Behind that sit the KPI definitions, and this is the part of the work that is easiest to underestimate. A process safety indicator is only as good as the rule that calculates it, and those rules are specific to the operator: what counts as an exceedance, how a partial period is treated, which events fall in scope. Encoding the client's definitions, and holding them consistent as scope widened, took a larger share of the effort than the interface did.

Scope grew in stages. The platform started with generation assets and later extended into LPG operations, which needed its own KPI set built on the same indicator framework. Governance dashboards were added to give oversight of the reporting process itself. Power BI integration was built so the data could feed the client's wider reporting estate instead of terminating in a tool of its own.

Component	What it does
Asset and Hazard Dashboard	Consolidated, at-a-glance indicator view tied to specific assets, from enterprise level down to individual sites
KPI definition layer	the client's own indicator rules encoded once and applied consistently across every view and asset class
LPG indicator set	Dedicated KPIs extending the same framework beyond the original generation-asset footprint
Governance dashboards	Oversight of the reporting process itself: completeness, currency and data quality
Power BI integration	Platform data made available to the client's broader reporting estate
Containerised deployment	Docker-based delivery running inside the client's environment and operated by the client's IT function

5. Why It Held Together

Plenty of internal reporting platforms get built and then quietly abandoned. This one did not, and the reasons are worth setting out, because they transfer.

The client held the pen

The client's program management reviewed scope line by line and made the calls on technical direction. That is more demanding for a supplier than being handed a specification, and it produces a better product, because the people who understand the operation are making the decisions that depend on understanding the operation.

Funding followed increments

Nobody had to commit to a four-year platform on day one. Each statement of work delivered something usable, which made the next one a straightforward case to make internally.

Scope grew because the platform earned it

LPG coverage, governance dashboards and Power BI integration were not in the original concept. They were added because the base platform was in daily use and people wanted more from it. That is a better signal than any satisfaction survey.

Ownership stayed with the client

The branding, the environment, the deployment model and the intellectual property: all the client's. A platform a client can run without its supplier is a platform that survives changes in supplier, budget and personnel. That was the intent from the start, and it is why the handover at the end of the build was a short conversation rather than a project of its own.

6. Outcomes

The plainest measure of the platform is that the client kept funding it. A platform backed across successive capital cycles over four years, extended into new asset classes and wired into the wider reporting estate, is a platform that is being used.

At completion in August 2026:

- Manual assembly of process safety reporting has been replaced by a single application drawing directly from source systems.
- Indicator definitions are held in one place and applied consistently, so the same question returns the same answer regardless of who asks it.
- Coverage spans generation and LPG operations under a common indicator framework.
- Platform data feeds the client's broader reporting through Power BI rather than sitting in isolation.
- The platform runs inside the client's environment, deployed and operated by the client's own IT function.
- Left AI-ready: built on a common architecture pattern with coding templates, and documented so AI coding assistants can extend it the same way.
- The build completed on schedule, with the final component delivered in August 2026 and support continuing through to November.

The client holds the underlying operational figures. This document does not publish numbers that are not ours to publish.

7. What This Means for Other Operators

The problem this platform solved is not particular to one operator. Any operator running high-hazard assets meets the same pattern: indicator data spread across systems, a manual reporting cycle that eats technical time, and a lag between something changing and anyone seeing it.

In Queensland that pressure has increased. Reporting obligations on generators have tightened, and critical control management expectations across resources have moved the same way. Operators are being asked to demonstrate that controls are working, on a cadence manual reporting struggles to sustain.

Three things from this engagement generalise.

Start with the definitions, not the dashboard

The interface is the visible part and the smaller part of the work. Agreement on how each indicator is calculated is what determines whether anyone trusts the output, and that agreement is harder to reach than it sounds. Get it settled early, in writing, with the people who will be held to the numbers.

Fund it in increments

Large reporting platforms specified up front tend to deliver what was imagined at the outset rather than what turned out to be needed. Smaller funded steps, each one usable on its own, produce a better result and are considerably easier to get approved.

Build for the client to own it

Client branding, client hosting, client deployment. If the supplier is a dependency, the platform has a shelf life measured in supplier relationships rather than in useful years.

8. About Succinct Software

Succinct Software is a Brisbane software consultancy, operating since 2009, working primarily with clients in the Australian energy and resources sectors.

The work covers legacy system modernisation, custom software development, business intelligence, process safety systems, integration and API development, and support after delivery.

Clients deal directly with the principal. There is no account management layer and no subcontracted delivery chain: the person who scopes the work architects it and stays with it. On complex systems that need to last, that continuity is the reason to engage a small firm rather than a large one.

Succinct has also delivered for AGL on hydro scheduling and optimisation, and for Vantage Point Systems in London, where a reporting and data aggregation platform delivered in 2012 has been continuously in service ever since.

C O N T A C T

Benjamin Snelling

Principal Consultant, Succinct Software Pty Ltd

benjamin@succinctsoftware.net

+61 468 941 444

succinctsoftware.net