Strategic Procurement and Feasibility Modelling for Large-Scale Autonomous Systems

Christopher I. V. Farmer, LL.M. (University of Worcester) · Independent Researcher · civfarmer@gmail.com

Preprint; not yet peer reviewed. Methodology paper. Version 1.1, October 2026.

Suggested citation: Christopher I. V. Farmer, 'Strategic Procurement and Feasibility Modelling for Large-Scale Autonomous Systems' (version 1.1, 2026) preprint. https://doi.org/10.5281/zenodo.23462150. ORCID 0009-0009-4995-2516. *Licence:* Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC 4.0). Commercial use needs a licence from the author: civfarmer@gmail.com.

> **Scope note.** This is a domain-general systems-engineering methodology paper. It presents transferable modelling techniques for estimating cost, schedule, and risk on very large, capital-intensive autonomous and infrastructure programmes — for example large renewable-energy build-outs, hyperscale data-centre programmes, autonomous logistics networks, and space infrastructure. The worked illustration is a generic, de-identified civilian autonomous-freight programme. All numerical values in the illustration are notional and clearly labelled as such; they are not empirical estimates for any real programme.

Abstract

Large-scale autonomous systems — networks that combine first-of-a-kind (FOAK) technology, heavy civil and energy infrastructure, and a novel software-and-sensing autonomy stack — are among the most difficult programmes to cost, schedule, and de-risk before commitment. The empirical record for megaprojects is unforgiving: cost overruns and schedule slippage are the norm, driven by optimism bias, immature technology, correlated risks, and long-lead supply constraints. This paper sets out an integrated, reproducible pre-commitment modelling methodology with five components: (1) a work-breakdown-structure (WBS)-driven capital-expenditure (CapEx) estimate carrying explicit uncertainty ranges and learning-curve effects for mass-produced elements; (2) critical-path scheduling under deep uncertainty using an event-sourced engine, *Helm*, that computes a working-calendar critical path, resource capacity, and Monte Carlo finish-date distributions and re-plans deterministically from a logged event stream; (3) feasibility *gates* — technology-readiness, supply-chain, power/energy budget, and integration risk — each with explicit go/no-go criteria; (4) a procurement layer covering make-versus-buy, single-source versus competitive tendering, and long-lead item management; and (5) an integrated feasibility index that triangulates bottom-up estimates against reference-class benchmarks. A fully de-identified worked illustration on a generic autonomous-freight network demonstrates the method end to end. The approach is technology-neutral and applies to any multi-billion-unit-of-currency autonomous or infrastructure programme.

Keywords: systems engineering; feasibility modelling; work breakdown structure; cost estimation; learning curve; technology readiness level; critical-path scheduling; Monte Carlo schedule risk; critical chain; procurement strategy; make-or-buy; long-lead procurement; megaproject; reference-class forecasting; go/no-go decision gates.

---

1. Introduction

A *large-scale autonomous system* couples three things at once: a fleet or network of many physical units capable of unsupervised or lightly-supervised operation; the fixed infrastructure (civil works, energy, communications) that hosts them; and an autonomy stack — sensing, compute, and decision software — that is frequently the least mature element in the whole undertaking. Such programmes are appearing across sectors: hyperscale and autonomously-operated data centres, gigawatt-scale renewable build-outs with automated balancing, autonomous logistics and freight networks, automated ports, and space infrastructure such as satellite constellations and their ground segments. They share scale (typically multi-billion in whatever currency), novelty (a FOAK or near-FOAK core), and a dependence on infrastructure and supply chains whose lead times dwarf the software development that usually attracts management attention.

The central problem this paper addresses is *feasibility modelling before commitment*: how to produce a defensible estimate of what such a programme will cost, how long it will take, and whether it can be delivered at all, at the point where the sponsor must decide whether to proceed, defer, or stop. This is precisely where the historical record is worst. The megaproject literature has established, across thousands of projects and several decades, that large capital programmes are delivered over budget and behind schedule "over and over again" (Flyvbjerg, 2014) — a pattern robust enough that Flyvbjerg named it, in the same article, the *iron law of megaprojects*, and later made it the theme of his introduction to the *Oxford Handbook of Megaproject Management* (Flyvbjerg, 2017). The causes are well characterised: *optimism bias* (systematic underestimation of cost, time, and risk), *strategic misrepresentation* (incentives to make projects look better than they are to win approval), immature technology carried into full-rate commitment, and the *correlation* of risks that naive estimates treat as independent.

Autonomous systems sharpen every one of these failure modes. The autonomy stack is typically at a low technology-readiness level (TRL) when budgets are set; the energy and grid-connection elements have lead times measured in years and are easy to under-scope; the sensing and compute supply chain is concentrated in a handful of vendors, creating single-source fragility; and the integration of novel software with novel physical infrastructure is a source of "unknown-unknowns" that point estimates do not capture.

The contribution of this paper is not a new theory but an *integrated, reproducible modelling method* binding five established bodies of practice — cost estimation, learning curves, schedule-risk analysis, technology-readiness assessment, and strategic procurement — into a single pre-commitment workflow, instrumenting the schedule with an auditable, event-sourced engine. The method is deliberately technology-neutral: nothing in it depends on what the autonomous units *do*; it models only money, time, maturity, supply, and risk. That neutrality is what makes it transferable across the sectors above.

The remainder of the paper is organised as follows. Section 2 reviews the related work the method draws on. Section 3 sets out the methodology in five layers, the fifth an integrating feasibility index. Section 4 works a fully de-identified illustration on a generic autonomous-freight network. Section 5 states the limitations. Section 6 concludes.

---

2. Related Work and Background

Systems-engineering lifecycle and the WBS. The methodology sits inside the standard systems-engineering (SE) lifecycle codified by the INCOSE *Systems Engineering Handbook* (INCOSE, 2023) and ISO/IEC/IEEE 15288:2023, both of which frame development as a progression through stages separated by *decision gates* at which the programme is assessed

and either continued or held. The *work breakdown structure* — a product-oriented, hierarchical decomposition of everything the programme must deliver — is the backbone artefact for cost and schedule estimation; the GAO lists it among the elements of a programme's technical baseline as the one that "identifies the cost and technical data needed to develop the estimate" (GAO, 2020a, p. 49). You cannot estimate, schedule, or de-risk what you have not decomposed.

Cost estimation and its discontents. The public-sector canon is the GAO *Cost Estimating and Assessment Guide* (GAO, 2020a), which defines the four characteristics of a reliable estimate (comprehensive, well-documented, accurate, credible; p. 31), insists on explicit *risk and uncertainty analysis*, a *range of confidence levels*, and adequate *contingency and management reserve*, and formalises *parametric estimating* (a statistical relationship between historical cost and a driver variable). The GAO warns that estimates carrying "meaningless confidence levels" are common when analysts do not understand the underlying uncertainty mathematics (GAO, 2020a, p. 13) — a caution this paper takes seriously.

Optimism bias and reference-class forecasting. The most influential empirical strand is Flyvbjerg's work on megaproject performance. Flyvbjerg, Holm and Buhl (2002) found systematic cost underestimation across hundreds of transport projects, with average real-terms overruns of roughly 45% for rail, 34% for fixed links (bridges and tunnels), and 20% for roads. Flyvbjerg (2006) proposed *reference-class forecasting* (RCF) as the corrective: rather than building a budget purely bottom-up, one identifies a class of comparable completed projects, derives the empirical distribution of their outcomes, and uses it to adjust the specific estimate. RCF operationalises Kahneman's *planning fallacy* (Lovallo & Kahneman, 2003; Kahneman, 2011) and was first put into practice in 2004, when the UK Department for Transport and HM Treasury adopted it to make the explicit, empirically based adjustments for optimism bias that the 2003 revision of HM Treasury's *Green Book* called for (Flyvbjerg, 2006); the current *Green Book* still requires appraisals to be adjusted for optimism bias (HM Treasury, 2026), with adjustment percentages set out in its supplementary guidance (HM Treasury, 2013). Flyvbjerg, Garbuio and Lovallo (2009) separated honest optimism from strategic misrepresentation; the *Oxford Olympics Study* (Flyvbjerg et al., 2016) showed some project classes carry average overruns well above 100%; and Flyvbjerg and Gardner (2023) synthesise the practical lessons. This paper uses RCF not as an alternative to bottom-up WBS estimation but as an independent *triangulation* against it.

Learning curves. For any programme with a mass-produced element, unit cost is not constant. Wright (1936), studying aircraft manufacturing, established that cost per unit falls by a roughly constant percentage with each doubling of cumulative output — the *learning* or *experience* curve, later generalised by the Boston Consulting Group (1970). Autonomous-systems programmes depend on exactly the components that display the steepest learning: photovoltaic module prices have fallen by about 27% (Fraunhofer ISE, 2026), and lithium-ion cell prices by about 20% (Ziegler & Trancik, 2021), with each doubling of cumulative output. Costing `N` units at the FOAK unit price is one of the largest single errors in FOAK estimation, and Section 3.2 treats it explicitly.

Technology readiness. Technology maturity is captured by the nine-level *Technology Readiness Level* (TRL) scale, originated at NASA and formalised by Mankins (1995), now standard across NASA, the U.S. Department of Defense and Department of Energy, and internationally; the GAO *Technology Readiness Assessment Guide* (GAO, 2020b) codifies best practice for assessing it. Two features matter here. First, this method takes a *system's* TRL to be the *lowest* TRL among its critical components — a conservative convention, since a system is no more mature than its least mature critical part. Second, acquisition practice expects a technology to have been demonstrated in a relevant environment — TRL 6 — before full development is committed. GAO observes that technologies typically pass into a formal acquisition programme at "at least a TRL 6 and preferably a TRL 7" (GAO, 2020b, p. 15). In the United States, a major defense acquisition program may not receive Milestone B approval, which opens engi-

neering and manufacturing development, until the milestone decision authority confirms that it considered, among other factors, that "the technology in the program or subprogram has been demonstrated in a relevant environment" (10 U.S.C. § 4252(b)(2), formerly § 2366b); DoD Instruction 5000.85 provides for that demonstration in the technology-maturation phase that precedes Milestone B (DoD, 2020). TRL is therefore a natural feasibility gate.

Scheduling under uncertainty. Network scheduling rests on two methods published in 1959: the deterministic critical-path method (CPM; Kelley & Walker, 1959) and the *Program Evaluation and Review Technique* (PERT; Malcolm et al., 1959), which expresses activity times in probability terms. Goldratt's (1997) *Critical Chain Project Management* reframed scheduling around the *Theory of Constraints*: estimate activities at their 50% (median) point, strip the padding hidden in individual tasks, and aggregate it into a shared *project buffer* sized to protect the delivery date, with *feeding buffers* protecting the critical chain from its tributaries. Because a single buffer size is hard to defend analytically, Hoel and Taylor (1999) showed Monte Carlo simulation can size it to a chosen completion probability, and Hulett (2011) developed *integrated cost-schedule risk analysis*, in which correlated Monte Carlo over the network yields a finish-date and cost distribution rather than a point. The *Helm* engine of Section 3.3 instantiates exactly this lineage — CPM backbone, critical-chain buffering against a committed date, Monte Carlo finish distribution — with an event-sourced architecture added for auditable re-planning.

Procurement strategy. Finally, the make-or-buy and sourcing decision is treated in the procurement and operations literature as a *strategic* rather than purely cost-driven choice, weighing unit economics against intellectual-property control, capacity, quality, and above all *supply risk*: single-source dependency, geographic and geopolitical concentration, and lead-time variability on long-lead items. The recommended posture — map single points of failure, then mitigate through dual-sourcing, localisation, inventory buffers, and design escrow — informs Section 3.5.

The ingredients are established; the contribution here is their *integration* into one pre-commitment feasibility model for autonomous systems, with the schedule instrumented by a reproducible engine.

---

3. Methodology

3.1 Overview: five layers and one decision

From a single WBS the method produces four quantitative views — a cost distribution, a schedule distribution, a maturity/feasibility assessment, and a procurement-risk posture — and combines them into one go/no-go recommendation. The five layers are:

1. **WBS-driven CapEx estimation** with three-point ranges, learning-curve treatment, and correlated aggregation (§3.2).
2. **Schedule modelling under deep uncertainty** with the *Helm* engine (§3.3).
3. **Feasibility gates** — TRL, supply-chain, energy/power budget, integration risk — each with an explicit pass/hold criterion (§3.4).
4. **Procurement strategy and risk** — make-vs-buy, sourcing, long-lead management (§3.5).
5. **An integrated feasibility index** that triangulates the bottom-up result against a reference class and rolls the gates into a single recommendation (§3.6).

The organising principle is *honest uncertainty*: every output is a range or distribution with a stated confidence level, never a single number presented as fact — the standard the GAO sets when it warns against "meaningless confidence levels."

3.2 WBS-driven CapEx estimation for first-of-a-kind systems

Decomposition. The estimate begins with a product-oriented WBS decomposed to the level at which a cost driver can be named for each element — typically WBS Level 3. For an autonomous system the top-level elements almost always separate into: the *fleet* (the replicated autonomous units); the *fixed infrastructure* (civil works, depots, sites); the *energy subsystem* (grid connection, substations, storage, charging or fuelling); the *autonomy stack* (sensing, compute, decision software — the FOAK core); the *orchestration/control layer*; *integration, verification and commissioning*; and *programme management and owner's costs*. The separation matters because these elements obey *different cost laws*: the fleet follows a learning curve, the infrastructure follows construction parametrics, and the autonomy stack behaves like R&D with a fat right tail.

Three-point ranges, not point estimates. Each terminal element receives a low (optimistic, ~P10), most-likely (mode), and high (pessimistic, ~P90) estimate that defines a distribution (triangular or PERT-beta). Mature, competitively-supplied commodities get tight ranges; the FOAK autonomy stack gets a deliberately wide, right-skewed range because its outcome is genuinely uncertain — and suppressing that skew is the modelling equivalent of optimism bias.

Learning curve for the replicated fleet. For the fleet element, costing `N` units at the FOAK unit price is a category error. Wright's law gives the cost of the `x`-th unit as

```
C(x) = C1 · x^(-b),   where   b = -ln(LR) / ln(2)
```

with `C1` the first-unit cost and `LR` the learning rate (e.g. `LR = 0.88` means a 12% cost reduction per doubling of cumulative output; `b > 0` whenever `LR < 1`). The *cumulative average* unit cost over `N` units is well approximated (for large `N`) by `C1 · N^(-b) / (1 - b)`, so total fleet cost is `N` times that. The gap between the naive estimate (`C1 · N`) and the learning-adjusted total is frequently a factor of two or more and can be the difference between a fleet line that looks unaffordable and one that is feasible. The learning rate itself is uncertain and should be carried as a range; the fleet cost distribution is then a function of both `C1` and `LR` distributions.

Parametric anchoring. Wherever a comparable database exists — construction cost per square metre, grid-connection cost per MW, storage cost per MWh, compute cost per rack — terminal elements are anchored to a *parametric* cost-estimating relationship (GAO, 2020a) rather than expert judgement alone. Parametrics keep the estimate connected to reality; expert three-point ranges handle the genuine novelty (the autonomy stack) where no parametric exists.

Correlated aggregation. The programme cost distribution is obtained by Monte Carlo aggregation of the element distributions — *not* by summing the most-likely values (which understates the mean because of skew) and *not* by assuming independence (which understates the tail). Element costs on one programme are positively correlated: the macro conditions that inflate one element — labour markets, commodity prices, schedule slippage — inflate others, so a modest assumed correlation (illustratively $\rho \approx 0.3$) materially fattens the aggregate tail and is far closer to reality than independence. From the aggregate the model reads the **P50** (the median outcome, as likely to be beaten as exceeded) and the **P80** (a prudent funding level); *contingency* is sized as roughly `P80 - P50` and attached to the risk register, and a *management reserve* of a few percent is held above P80 for unknown-unknowns, per GAO practice.

Triangulation against a reference class. The bottom-up P80 is finally cross-checked against a *reference-class* estimate (Flyvbjerg, 2006): take a comparable class of completed FOAK programmes and apply their empirical *optimism-bias uplift* to the naive point estimate. Agreement raises confidence; a bottom-up number far below the reference class signals optimism and sends the decomposition back for revision. The two methods are complementary — bottom-up for structure, reference-class for realism.

3.3 Schedule modelling under deep uncertainty: the *Helm* engine

Cost cannot be separated from time: on capital programmes, schedule slip *is* cost, through prolonged overheads, financing, and the escalation of un-let contracts. The schedule layer therefore uses the author's project-scheduling engine, **Helm**, which is designed for planning under deep uncertainty and, importantly, for *auditable re-planning* as the programme moves.

Event-sourced core. Helm's central abstraction is a pure function

```
plan = schedule(seed, events, today)
```

The **seed** is an immutable description of the intended programme: *tasks* (each with an id, duration, dependency list, assigned resource, and low/medium/high risk class), *resources* (each with a full-time-equivalent capacity and internal/external flag), optional *stages*, a committed *deadline*, and a fixed *Monte Carlo seed*. The **events** are an append-only log of what has actually happened or been decided — delays, blockers, progress updates, duration changes, reassignments, task splits, "not-before" constraints, decisions. The **plan** is a deterministic function of the two, which gives the architecture its two defining properties: it is *reproducible* (same seed + events + Monte Carlo seed ⇒ identical outputs, so any number in a board paper regenerates exactly) and *auditable* (every deviation from baseline is carried in the event that caused it).

Dependency graph with overlap. Dependencies are not restricted to finish-to-start; a dependency may be *fractional* — "start when the predecessor is `p` fraction complete" — letting the model represent deliberate *overlap* (fast-tracking), e.g. beginning software integration when the fleet build is 40% delivered rather than waiting for the last unit. Fast-tracking is one of the few genuine levers for schedule compression on a deadline-constrained programme, and modelling it explicitly rather than by hand-shortening durations keeps the compression honest and visible.

Deterministic backbone. Over a *working calendar* (weekends removed), Helm computes a forward pass giving each task a start day and end day, and a backward pass from the committed deadline giving each task its *slack*, counted in calendar days; tasks with no slack left form the **critical path**, and tasks within three days of the least slack are flagged as *driving*. This is standard CPM, made calendar-honest.

Resource capacity. Because durations assume resources are available, Helm levels each internal resource — one task at a time, at its full-time-equivalent capacity — and reports its *capacity* utilisation across a window, which prevents the classic error of a schedule that is feasible task-by-task but infeasible in aggregate because one scarce resource is claimed by three parallel tasks at once. External lanes, such as a vendor or grid contractor, are queues whose durations are taken as given: Helm does not level them, so where one is overcommitted the apparent critical path is optimistic.

Monte Carlo finish distribution. The deterministic critical path is the *optimistic* schedule. Helm perturbs task durations by risk class and re-runs the network many times against its seeded generator, producing a *distribution of finish dates* from which it reports a **P80 finish** (met in 80% of simulations) and a **confidence** figure (the share of runs that meet the deadline). Its **buffer**, by contrast, is deterministic: the margin, in calendar days, between the deterministic finish and the deadline — Goldratt's critical-chain project buffer. Sizing that buffer by Hoel–Taylor Monte Carlo rather than rule of thumb, and applying Hulett-style integrated risk to the finish date, is then a reading of the distribution: at 80% confidence the risk consumes the stretch from the deterministic finish to the P80, and what is left between the P80 and the deadline is the **P80 margin**, which Helm does not report.

Feasibility and health. Helm rolls the confidence into a **feasibility label** — *Viable* at 85% or more, *Viable but tight* from 60%, *Viable only with reduced scope* from 42%, *Viable only with more resource* from 28%, and *Not viable under current constraints* below that — and sets a

traffic-light **health** state: green when confidence is at least 80% and no notice or escalation (below) is open; amber when confidence is below 80% or a notice or escalation is open; red when any escalation is red; black when the deterministic finish passes the deadline by more than five days. A confidence below 80% means the P80 has already passed the deadline, so on that ground alone Helm shows amber, not red. The method reads the P80 margin more strictly: a thin margin is a warning, and a P80 past the deadline is a breach that calls for a decision, whatever Helm's health shows (§3.6).

Evidence-based progress. The engine enforces one principle: *progress is only ever recognised from a logged event*, never inferred from a stage label or a plan that "should" have advanced. Nothing is marked complete by assumption — the scheduling analogue of earned-value discipline, which directly resists the optimism bias that makes reported progress outrun real progress.

Logged-delay re-planning. When something slips, the response is not to redraw the chart by hand but to *append an event* — a delay, a blocker with an impact in days, a duration revision. Re-running `schedule` re-levels the entire network: downstream tasks move, the critical path may switch, and the Monte Carlo distribution, buffer, and health state recompute. Because the causing event carries a note, the *reason* travels with the moved task, so a reviewer sees not just that the finish date moved but *why*. Helm also compares the current plan against the **baseline** (the same seed with an empty event log), reporting drift explicitly: how many tasks have moved and how far the finish date has slipped from the approved plan.

Decision support: escalations. Finally, Helm surfaces *escalations* structured as *what happened / what was tried / the impact / the options*, so that the reviewer is handed a decision rather than a bare red flag. Helm raises them on explicit rules: a blocker on a critical-path task, a deterministic finish past the deadline, a confidence below 60%, a decision falling due, a call for help on a driving task, or a dedicated resource left idle. A blocker on a task that still has slack is logged instead as a *notice* that the slack absorbs it: health turns amber, but nothing is asked of the reviewer. Apart from decisions the planner has set up in advance, Helm's options are generic — chase the blocking party, re-scope, add resource, negotiate the deadline — so programme-specific options, such as interim on-site generation or phased commissioning for a delayed energisation, are the analyst's to frame; the method frames them in the same structure for a P80 breach that Helm does not escalate. Together they turn the schedule model from a reporting tool into a decision instrument.

3.4 Feasibility gates and go/no-go criteria

Cost and schedule distributions answer "how much" and "how long"; *feasibility gates* answer "can this be done at all, now?" Each gate is a graded test with an explicit criterion; failing one does not cancel a programme but bars it from full-rate commitment until cleared — usually via a dedicated maturation phase. The method uses four gates, aligned with the stage-gate discipline of Cooper (1990) and the decision gates of the SE lifecycle (INCOSE, 2023).

Gate 1 — Technology readiness. Assess the TRL of every critical subsystem (Mankins, 1995; GAO, 2020b); the method takes the *system* TRL to be the minimum across them. Criterion: the system TRL must reach the acquisition threshold — conventionally **TRL 6**, demonstrated in a relevant environment (10 U.S.C. § 4252; DoD, 2020; GAO, 2020b) — before committing to build. The *pacing item* is the lowest-TRL critical subsystem, and for autonomous systems it is almost always the autonomy decision stack or whole-system integration. A sub-threshold TRL sends the programme to a technology-maturation phase rather than to commitment.

Gate 2 — Supply chain. Assess, for every critical procured item, the number of qualified suppliers, the geographic/geopolitical concentration, and the lead time and its variability. Criterion: no *critical single point of failure* on the delivery path without a funded mitigation (dual-source, buffer stock, design escrow). Concentrated, long-lead items (advanced compute,

high-power grid transformers, battery cells) fail this gate unless mitigated.

Gate 3 — Power/energy budget. Autonomous systems are energy systems. Compute the aggregate energy and, critically, the *peak power* demand, and test it against what candidate sites and grid connections can deliver within the schedule. Criterion: firm capacity — including the grid-connection *lead time* — must be securable within the plan, or the deficit closed by on-site generation/storage. This gate frequently binds harder than any software risk, because grid connection is both a long-lead item and a hard physical constraint.

Gate 4 — Integration risk. Assess the number and novelty of interfaces between the FOAK autonomy stack and the physical infrastructure, and the whole-system integration TRL. Criterion: integration maturity must be demonstrated (typically via a representative prototype/pilot) before full-rate build; a low integration TRL with many novel interfaces is the classic source of unknown-unknowns and fails the gate.

The gates are *conjunctive*: the programme is feasible for commitment only if all four pass. A single red gate is a *no-go for commitment* and a *go for maturation* — the correct, and often politically difficult, outcome that the method is designed to make defensible.

3.5 Procurement strategy and risk

Procurement is where the cost, schedule, and supply-risk pictures become contractual commitments, and its choices feed straight back into the model.

Make versus buy. Each major element's make/buy choice is decided on *strategic* grounds, not unit cost alone (see §2). The heuristic: *buy* mature, competitively-supplied elements (the hardware platform, commodity sensors) to capture market pricing and suppliers' already-realised learning; *make or partner-with-control* the core-IP differentiators (the autonomy decision stack, the orchestration layer), because control of the FOAK core is the programme's actual value and cannot be safely outsourced. Where a strategic element must be bought (few suppliers exist), the buy is protected with *design/source-code escrow* and second-source qualification so a supplier failure is survivable.

Single-source versus competitive. Competitive tendering disciplines price and improves resilience, and is the default wherever a real supplier market exists. But genuine single-source situations are common — advanced compute and specialised sensors may have only one or two viable suppliers. Single-source is not automatically wrong (it can buy integration coherence and priority in an allocation-constrained market) but must be *recognised and mitigated* through buffer stock, an invested second-source, and contractual protection. The model records each critical item's sourcing posture and residual risk, feeding Gate 2.

Long-lead items. The items that most often determine feasibility have the longest, least-compressible lead times — grid connections and high-power transformers, battery cells, advanced compute. The rule is explicit: *identify long-lead items first, and reserve them ahead of the works that logically precede them.* In Helm these appear as long-duration tasks carrying *not-before* constraints, almost always on or near the critical path; their lead time, not the construction or software, is frequently the true schedule driver. Committing early — before design is complete — trades a bounded financial risk (a reservation that may be wasted) against an unbounded schedule risk (a multi-year connection queue). For deadline-constrained programmes that trade is usually correct, and the model quantifies it: the *cost of early commitment* against the *schedule value of the buffer it protects*, read off the Helm buffer and the cost distribution.

3.6 Bringing it together: an integrated feasibility view

The four quantitative views are combined into a single recommendation:

- **Cost:** fund at the aggregate **P80**, hold **management reserve** above it, and confirm the bottom-up P80 is consistent with the **reference-class** benchmark (§3.2).
- **Schedule:** commit to a date at or beyond the Helm **P80 finish**, so that the **P80 margin** is non-negative, with an acceptable **confidence**; health must be green or a well-understood amber (§3.3).
- **Gates:** all four feasibility gates pass, or a costed and scheduled **maturation phase** is defined to clear the failing gate(s) (§3.4).
- **Procurement:** every critical item has a defined sourcing posture with residual risk mitigated to acceptable, and long-lead commitments are sequenced (§3.5).

A programme satisfying all four is a *go*. One that fails on affordability, on a schedule whose P80 passes the committed date, or on a gate that cannot be cleared within the sponsor's constraints is a *no-go-for-now* with a defined path (usually maturation) to a future gate. The method's value is that this recommendation is *reproducible and auditable*: every number regenerates from the WBS, the estimate assumptions, and the Helm seed-plus-events, so the decision can be defended and, when circumstances change, cheaply re-run.

---

4. Worked Illustration

> **All figures in this section are illustrative and notional.** They are chosen to demonstrate the method and its arithmetic on a realistic-looking but entirely generic programme. They are not estimates for any actual project, product, or organisation, and no figure should be cited as an empirical result.

The programme. Consider *Programme A*, a hypothetical FOAK national **autonomous electric-freight shuttle network**: roughly 2,000 self-driving electric cargo shuttles on a defined route network, served by twelve automated depots/terminals, an energy and charging backbone, an autonomy sensing-and-decision stack, and a central orchestration ("control tower") layer. Programme A is representative only; the identical modelling applies to a renewable build-out, a hyperscale data-centre, or a satellite constellation by re-labelling the WBS.

4.1 WBS and CapEx estimate

Table 1 gives the illustrative Level-1 WBS with three-point CapEx ranges (currency units, "cu", in millions).

Table 1 — Illustrative WBS CapEx (cu millions, notional).

WBS	Element	Low	Most-likely	High	Cost law
1.0	Autonomous vehicle fleet (2,000 units)	520	600	780	Learning curve
2.0	Automated depots & terminals (×12)	700	820	1,150	Construction parametric
3.0	Energy & charging backbone	520	640	950	Parametric + grid
4.0	Autonomy & sensing stack (FOAK core)	380	520	980	R&D, right-skewed
5.0	Network control-tower / orchestration	175	240	420	Software
6.0	Integration, V&V & commissioning	220	300	560	% of build, risk-loaded
7.0	Programme mgmt & owner's costs	285	330	470	% of programme
	Sum of most-likely		**3,450**		

Learning-curve derivation of the fleet line (1.0). Naively costing 2,000 shuttles at an illustrative FOAK unit price `C1 = 0.90` cu gives `0.90 × 2,000 = 1,800` cu — the largest line in the programme. Applying Wright's law with an illustrative learning rate `LR = 0.88` (`b = −ln 0.88 / ln 2 ≈ 0.184`), the cumulative-average unit cost over 2,000 units is `C1 · N^(−b) / (1 − b) ≈ 0.272` cu, so the total build ≈ `2,000 × 0.272 ≈ 543` cu; with tooling and spares the fleet most-likely is ≈ **600 cu**, one-third of the naive figure. The learning curve is the single largest correction in the estimate, and treating it explicitly is what makes the fleet line credible. The row's low and high values (520 and 780 cu) carry the uncertainty in `C1` and `LR`: at the same `C1` and tooling-and-spares allowance, they correspond to learning rates of about 0.865 and 0.91.

Aggregate distribution. Summing the most-likely column gives 3,450 cu — neither the mean nor a safe budget. Monte Carlo aggregation of the seven element distributions under a modest positive correlation (`ρ ≈ 0.3`) yields an illustrative **P50 ≈ 3,940 cu** and **P80 ≈ 4,490 cu** (mean ≈ 3,970 cu). Each element is modelled as triangular, with its most-likely value as the mode and its bounds set so that its low and high values in Table 1 are its 10th and 90th percentiles (§3.2); the elements are joined by a Gaussian copula with a uniform correlation of 0.3, and the figures come from 200,000 draws with a fixed seed (2026). The method funds at ≈ 4,490 cu, with *contingency ≈ P80 − P50 ≈ 550 cu* on the risk register and a *management reserve of ~5% (~225 cu)* above P80 — a total authorisation of about 4,715 cu. Two traps are avoided: summing most-likely values under-funds by ~1,040 cu against P80, and assuming independence would have produced a falsely tight P80 (≈ 4,290 cu).

Reference-class triangulation. Independently applying an illustrative 30% optimism-bias uplift (drawn from a reference class of comparable FOAK programmes) to the naive estimate gives `3,450 × 1.30 ≈ 4,485 cu`. That the top-down figure (~4,485 cu) and the bottom-up P80-plus-reserve (~4,490–4,715 cu) *land in the same region* is the signal the method seeks: the estimate is internally consistent and not obviously optimistic.

4.2 Technology-readiness assessment (Gate 1)

Table 2 — Illustrative TRL assessment.

Subsystem	Illustrative TRL	Note
Vehicle platform (drive-by-wire EV)	8	Mature, adapted
Energy & charging backbone	8	Established
Depot automation / mechatronics	7	Near-mature
Perception / sensor suite	6	At threshold
Fleet orchestration software	6	At threshold
Autonomy decision stack (full self-driving in ODD)	**5**	**Pacing item**
Whole-system integration	**4**	**Pacing item**

The **system TRL is 4** (the minimum). Against the conventional **TRL 6** commitment threshold, **Gate 1 does not pass**: the autonomy decision stack (TRL 5) and whole-system integration (TRL 4) are immature. The correct decision is *no-go for full commitment, go for a technology-maturation phase* that raises the autonomy stack and the integration to TRL 6 via a representative prototype/pilot before the main build is authorised.

4.3 Power/energy budget (Gate 3)

Illustratively, 2,000 shuttles averaging 250 kWh/day consume ≈ **500 MWh/day** (≈ 150 GWh/year over ~300 operating days). The binding number is *peak power*: if 40% of the fleet charges concurrently at 150 kW, peak demand is `2,000 × 0.40 × 150 kW = 120 MW`, ≈ 10 MW at each of the twelve depots. **Gate 3** therefore requires securing ≈120 MW of firm grid capacity *and* accommodating the connection lead time within the schedule (§§4.4–4.5); where firm capacity is unavailable in time, the deficit is closed with on-site storage and staged energisation — a design and cost input, not an afterthought.

4.4 Schedule model in Helm

Table 3 gives the illustrative Helm seed (durations in working days; dependencies show fractional overlaps and not-before days where used). Working days are converted to months at 20.8 a month (about 250 a year); on Helm's calendar of five working days a week, working day 998 (month 48 on this convention) is calendar day 1,396, about 46 calendar months after the start. Following §3.5, the two long-lead tasks, D and E, are reserved at concept freeze, ahead of the maturation and detailed design that would logically precede them.

Table 3 — Illustrative Helm schedule seed.

Task	Description	Dur (wd)	Depends on	Resource
A	Concept & requirements	40	—	Engineering
B	Technology maturation & prototype demo	120	A	Autonomy team
C	Detailed design	90	B	Engineering
D	Grid connection & energisation *(long-lead)*	520	A (not-before day 40)	Grid contractor (ext.)
E	Autonomy compute & sensor procurement *(long-lead)*	300	A (not-before day 40)	Vendor (ext.)
F	Depot civil works	180	C	Civils (ext.)

Task	Description	Dur (wd)	Depends on	Resource
G	Energy & charging installation	120	D, F@0.7	Civils (ext.)
H	Fleet build ramp	220	C	Vendor (ext.)
I	Autonomy software integration & V&V	160	E, H@0.4	Autonomy team
J	System integration & commissioning	90	G, I	Integration/T&E
K	Pilot operations & acceptance	60	J	Operations

Deterministic result. The longest chain is `A → D → G → J → K` (`40+520+120+90+60 = 830` working days ≈ **month 40**), *driven by the grid-connection long-lead task D*, not the software, even though D is ordered at concept freeze; the autonomy chain via E (650 wd) and the fleet chain via H (648 wd with the 40% overlap) run inside it, with about 180 working days of float. This is counter-intuitive but characteristic: on an autonomous-systems programme the pacing schedule item is frequently the *energy/grid* element, because its lead time is the least compressible. Ordering D and E only after detailed design (C) — the sequencing §3.5 warns against — would put the finish at 1,040 working days (≈ month 50): reserving the long-lead items early is worth 210 working days (≈ 10 months) here.

Monte Carlo result. Perturbing durations by risk class — every task here carries the medium class, Helm's default: a uniform factor of 0.75–1.27 on its duration, scaled by a common per-run factor of 0.93–1.12, with a 3.5% chance of a 1.6× overrun — Helm's seeded simulation (400 runs, Monte Carlo seed 42) yields an illustrative **P80 finish ≈ month 47** (970 wd). Against a committed deadline of **month 48** (998 wd), Helm reports a **confidence of 87%**, a **buffer of 236 days** (168 working days beyond the deterministic finish), the label *Viable* and a **health state of green**. The method's reading is less comfortable: at 80% confidence the risk consumes 140 of those 168 working days, leaving a **P80 margin of only 28 working days (≈ 1.3 months)** — deliverable, but thin, and dominated by a single long-lead item.

Logged-delay re-planning. Suppose grid energisation (task D) subsequently slips: the planner appends a `blocker` event of +60 working days noted "network-operator connection-queue delay." Re-running `schedule` re-levels the network: D, G, J and K each move 60 working days later, the baseline comparison reports the drift explicitly, and the P80 finish moves out by 60 working days to ≈ **month 49.5** (1,030 wd), past the deadline by 32 working days (≈ 1.5 months). Helm's buffer, measured from the deterministic finish (now 890 wd), falls from 236 to 152 days; confidence falls to 72%, the label becomes *Viable but tight*, and health turns **amber**. Helm raises **no escalation**: D still has 158 days of slack against the deadline, so the blocker is logged as a notice — "absorbed by 158d of slack; downstream tasks re-sequenced, no gate impact" — and confidence stays above the 60% threshold.

Escalation. Read by the method, the same event is a deadline breach, which the analyst frames in Helm's structure — *what:* grid energisation delayed 60 wd; *tried:* expedite request; *impact:* +≈3 months to P80 finish (≈ 1.5-month deadline breach); *options:* (a) interim on-site generation to decouple pilot start from full energisation, (b) phased depot energisation to bring part of the network live earlier. The decision-maker is handed a *decision*, not an amber light — which is why §3.6 commits on the P80 margin and not on Helm's health alone.

4.5 Procurement posture (Gates 2, 4 and strategy)

Table 4 — Illustrative procurement and long-lead posture.

Item	Make/Buy	Sourcing	Illustrative lead	Mitigation
Vehicle platform	Buy	Competitive	9–12 mo	Market tender; captures supplier learning
Autonomy decision stack	Make / partner-with-control	Strategic	n/a (core IP)	In-house control; source-code escrow if partnered
Autonomy compute	Buy	**Single/dual-source**	9–18 mo	Buffer stock; qualify second source
Sensor suite	Buy	Competitive/dual	6–12 mo	Dual-source; standard interfaces
Battery cells	Buy	Few-source	12–18 mo	Advance reservation; buffer stock
Grid connection / transformers	Buy	**Single (network operator)**	**18–30 mo**	**Order first (task D, not-before day 40); interim generation option**
Orchestration software	Make	Strategic	n/a	Core IP retained

Gate 2 (supply chain) does not pass unmitigated: grid connection and advanced compute are concentrated, long-lead single points of failure. With the mitigations shown — early ordering, buffer stock, second-source qualification — residual risk is acceptable and the gate passes *conditionally on those mitigations being funded and scheduled*. **Gate 4 (integration)** fails at system TRL 4 and, like Gate 1, is cleared only by the maturation phase (task B and the pilot). The long-lead grid connection is deliberately sequenced first — a bounded early commitment bought to protect the otherwise unbounded 18–30-month schedule risk that §4.4 shows to be the true critical driver.

4.6 Integrated recommendation

Rolling the layers together for Programme A: fund at ≈ P80 + reserve (≈ 4,715 cu), consistent with the reference class; commit no earlier than the Helm P80 (month 47–48), recognising that the P80 margin is thin and grid-dominated although Helm reports green; **but hold full commitment** because Gates 1 and 4 fail at system TRL 4. The defensible decision is a *staged go*: authorise a costed technology-maturation phase (raise autonomy and integration to TRL 6, run the pilot), place the long-lead grid and compute commitments now, and defer full-rate build authorisation to a second gate once maturity and schedule margin are demonstrated. Every figure regenerates from the WBS, the estimate assumptions and the Helm seed-plus-events stated above — which is what lets the staged decision be defended and cheaply re-run as conditions change.

---

5. Limitations

Garbage-in remains garbage-in. The method disciplines and makes transparent the assumptions behind a feasibility case; it does not manufacture data. Three-point ranges, learning rates, correlations, and risk classes are judgements, and a biased analyst can still produce

a biased model. Reference-class triangulation is the main guard, but it depends on a genuinely comparable class — which, for the most novel FOAK programmes, may be thin or absent, leaving the estimate resting more heavily on expert judgement whose uncertainty should then be widened.

Correlation is hard, and TRL is coarse. The aggregate cost tail is sensitive to the assumed inter-element correlation, which is difficult to estimate; the method's modest-positive default is defensible but is an assumption, and the P80's sensitivity to it should be reported. Likewise the nine-level TRL scale is a useful common language but a blunt instrument — it compresses many risks into one integer, can be gamed, and says little about *integration* specifically. Treating integration as its own gate compensates only partly.

Monte Carlo is only as good as its inputs. Helm's finish-date distribution depends on the risk-class duration model and on the seed's dependency structure; missing dependencies or unmodelled resource contention make any schedule optimistic. *Deep* uncertainty — the model being structurally wrong, not merely its parameters uncertain — is captured by no Monte Carlo, which is why the management reserve and staged-commitment posture, not the distribution itself, are the real guard.

Illustration, not evidence; governance, not modelling. The Section 4 numbers are notional throughout, chosen for pedagogical clarity, not fitted to data. And strategic misrepresentation (Flyvbjerg et al., 2009) is a *governance* problem: an auditable model raises the cost of misrepresentation by making assumptions explicit but cannot remove the incentives that produce it — that requires independent review and accountable decision gates around the model.

---

6. Conclusion

Large-scale autonomous systems fail at the feasibility stage for reasons now well understood: optimism about cost and schedule, immature technology carried too far too fast, concentrated and slow supply chains, and correlated risks treated as independent. None of the individual remedies is new; the contribution here is to bind five of them — WBS-driven three-point estimation with learning curves, correlated Monte Carlo aggregation triangulated against a reference class, TRL-and-integration feasibility gates, event-sourced schedule-risk modelling in the *Helm* engine, and a procurement posture centred on long-lead items — into one reproducible, auditable, pre-commitment workflow, shown end-to-end on a generic autonomous programme. The discipline is honest uncertainty: fund at a stated confidence level, commit only past demonstrated maturity, and let every number be regenerated from its inputs so a hard "no-go-for-now, here is the path" can be defended as readily as an easy "go." The approach is technology-neutral and transfers directly to renewable build-outs, hyperscale data centres, autonomous logistics, and space infrastructure — any programme large enough, novel enough, and slow enough that guessing is not good enough.

References

Boston Consulting Group. (1970). *Perspectives on Experience.* Boston, MA: BCG.

Cooper, R. G. (1990). Stage-Gate Systems: A New Tool for Managing New Products. *Business Horizons, 33*(3), 44–54.

Flyvbjerg, B., Holm, M. S., & Buhl, S. (2002). Underestimating Costs in Public Works Projects: Error or Lie? *Journal of the American Planning Association, 68*(3), 279–295.

Flyvbjerg, B. (2006). From Nobel Prize to Project Management: Getting Risks Right. *Project Management Journal, 37*(3), 5–15.

Flyvbjerg, B., Garbuio, M., & Lovallo, D. (2009). Delusion and Deception in Large Infrastructure Projects: Two Models for Explaining and Preventing Executive Disaster. *California Management Review, 51*(2), 170–193.

Flyvbjerg, B. (2014). What You Should Know About Megaprojects and Why: An Overview. *Project Management Journal, 45*(2), 6–19.

Flyvbjerg, B., Stewart, A., & Budzier, A. (2016). *The Oxford Olympics Study 2016: Cost and Cost Overrun at the Games.* Saïd Business School Working Papers, University of Oxford (arXiv:1607.04484).

Flyvbjerg, B. (2017). Introduction: The Iron Law of Megaproject Management. In B. Flyvbjerg (Ed.), *The Oxford Handbook of Megaproject Management* (pp. 1–18). Oxford: Oxford University Press.

Flyvbjerg, B., & Gardner, D. (2023). *How Big Things Get Done.* New York: Currency.

Fraunhofer Institute for Solar Energy Systems ISE. (2026). *Photovoltaics Report* (updated 14 July 2026). Freiburg: Fraunhofer ISE.

Goldratt, E. M. (1997). *Critical Chain.* Great Barrington, MA: North River Press.

HM Treasury. (2013). *Supplementary Green Book Guidance: Optimism Bias.* London: HM Treasury.

HM Treasury. (2026). *The Green Book: UK Government Guidance on Appraisal.* London: HM Treasury.

Hoel, K., & Taylor, S. G. (1999). Quantifying Buffers for Project Schedules. *Production and Inventory Management Journal, 40*(2), 43–47.

Hulett, D. T. (2011). *Integrated Cost-Schedule Risk Analysis.* Farnham: Gower.

INCOSE. (2023). *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities* (5th ed.; D. D. Walden et al., Eds.). Hoboken, NJ: Wiley.

ISO/IEC/IEEE. (2023). *ISO/IEC/IEEE 15288:2023 — Systems and Software Engineering — System Life Cycle Processes* (2nd ed.). Geneva: International Organization for Standardization.

Kahneman, D. (2011). *Thinking, Fast and Slow.* New York: Farrar, Straus and Giroux.

Kelley, J. E., Jr., & Walker, M. R. (1959). Critical-Path Planning and Scheduling. In *Papers Presented at the December 1–3, 1959, Eastern Joint IRE-AIEE-ACM Computer Conference* (pp. 160–173). New York: ACM.

Lovallo, D., & Kahneman, D. (2003). Delusions of Success: How Optimism Undermines Executives' Decisions. *Harvard Business Review, 81*(7), 56–63.

Malcolm, D. G., Roseboom, J. H., Clark, C. E., & Fazar, W. (1959). Application of a Technique for Research and Development Program Evaluation. *Operations Research, 7*(5), 646–669.

Mankins, J. C. (1995). *Technology Readiness Levels: A White Paper.* Washington, DC: NASA, Office of Space Access and Technology.

10 U.S.C. § 4252. Major defense acquisition programs: Factors to be considered before Milestone B approval (formerly § 2366b; as amended by Pub. L. 118-159, § 807, 2024, and Pub. L. 119-60, § 1811(g)(3), 2025).

U.S. Department of Defense. (2020). *DoD Instruction 5000.85: Major Capability Acquisition* (6 August 2020, incorporating Change 1, 4 November 2021). Washington, DC: DoD.

U.S. Government Accountability Office. (2020a). *Cost Estimating and Assessment Guide* (GAO-20-195G). Washington, DC: GAO.

U.S. Government Accountability Office. (2020b). *Technology Readiness Assessment Guide* (GAO-20-48G). Washington, DC: GAO.

Wright, T. P. (1936). Factors Affecting the Cost of Airplanes. *Journal of the Aeronautical Sciences, 3*(4), 122–128.

Ziegler, M. S., & Trancik, J. E. (2021). Re-examining Rates of Lithium-Ion Battery Technology Improvement and Cost Decline. *Energy & Environmental Science, 14*(4), 1635–1651.

Preprint; not yet peer reviewed. All numerical values in the worked illustration are illustrative and notional. This paper contains no weapons, munitions, or operational-military content of any kind.

Further reading and points of disagreement (July 2026 addendum)

The worked illustration in this paper assumes learning-curve improvement and stable procurement incentives; both assumptions have a critical literature that a careful reader should weigh. The empirical learning-curve record since Wright shows large variance in realised learning rates across organisations and programmes — Argote and Epple's survey is the standard corrective to any deterministic projection. The procurement-incentives literature, exemplified by Rogerson, analyses the incentive problems between government and defence firms, and within government itself, that shape defence procurement; and the base rates assembled by Flyvbjerg for megaprojects, and by Merrow for industrial megaprojects and — with colleagues at RAND — for pioneer (first-of-a-kind) process plants, quantify how far large programmes drift from their estimates. Read against that canon, the appropriately cautious position — and the one this paper intends — is that the illustration's numbers are notional throughout, the feasibility claim is confined to structure rather than point estimates, and any real programme would govern commitment through staged technology-readiness gates in Mankins' sense rather than through the illustration's smooth curves.

- Argote, L. & Epple, D., "Learning curves in manufacturing" (1990) 247 *Science* 920–924. https://doi.org/10.1126/science.247.4945.920.
- Rogerson, W. P., "Economic incentives and the defense procurement process" (1994) 8 *Journal of Economic Perspectives* 65–90. https://doi.org/10.1257/jep.8.4.65.
- Flyvbjerg, B., "What you should know about megaprojects and why: an overview" (2014) 45 *Project Management Journal* 6–19. https://doi.org/10.1002/pmj.21409.
- Merrow, E. W., *Industrial Megaprojects: Concepts, Strategies, and Practices for Success* (Wiley 2011).
- Merrow, E. W., Phillips, K. & Myers, C. W., *Understanding Cost Growth and Performance Shortfalls in Pioneer Process Plants* (RAND R-2569-DOE, 1981). https://www.rand.org/pubs/reports/R2569.html.
- Mankins, J. C., "Technology readiness assessments: a retrospective" (2009) 65 *Acta Astronautica* 1216–1223. https://doi.org/10.1016/j.actaastro.2009.03.058.