Chapter 1: Why Anyone Pays for a Digital Twin

Before You Start

What you are assumed to know. You write software for a living. You are comfortable with HTTP APIs, JSON, a time-series or relational database, and reading Python you did not write. You have seen a system that talks to hardware, even if you did not build it.

What you are not assumed to know. Anything about control theory, numerical simulation, process chemistry, or the digital-twin literature. Where this chapter needs a term from one of those fields, it defines the term at first use and then keeps using it.

What this chapter covers. Why organisations spend money on digital twins; how that spending is justified before a line of code exists; what the published market numbers do and do not tell you; what six recurring shapes of value look like and where the human stands in each; where the cost sits; how four families of real deployments argue for their money; and how to build the value case for a small system, worked in full on a reactor vessel that is dosed with reagent on a timer.

What this chapter deliberately does not cover. What a digital twin *is*, technically. That sounds like the wrong order, and it is a deliberate choice. A later chapter draws the line between a twin, a shadow, a model and a simulation, and another gives the architecture. Both are easier to follow once you know what the thing is *for*. Until then, this chapter runs on one working sentence:

> **Digital twin (working definition, Part I only).** Software that keeps a model of one specific physical system in step with that system's actual state, and uses it to control that system and to answer questions about it that measurement alone cannot answer.

That sentence is deliberately loose. The literature does not agree on where a *digital twin* ends and a *digital shadow* begins, and a shared, transparent understanding of the two terms is still being argued for in manufacturing [1]. The tightening matters, because the boundary carries cost consequences, and a later chapter does it. For now the loose version is enough to talk about money.

Two more terms, pinned now and used unchanged for the rest of the book. The **physical twin (PT)** is the real system being twinned. In the worked example below that is the reactor vessel, its contents, the concentration sensor, the dosing pump, the tubing, and the controller that samples the sensors. The **digital twin (DT)** is the software defined above. From here on this chapter writes PT and DT rather than spelling either out, and so does the rest of the book. When the book says "the twin" with no qualifier, it means the digital one.

Learning objectives. By the end of this chapter you will be able to:

1. **Name the decision** a proposed twin is meant to improve, and turn a vague goal into one value metric with a unit, a denominator, a measurement method and a measured baseline.
2. **Classify** a proposed or existing system into one of six value patterns, state which rung of the ladder it sits on, and say what the next rung up would additionally require, including who acts on its output.

3. **Rank** the cost lines of a first-increment twin for a given system, and identify which line is most likely to be forgotten.
4. **Critique** a market forecast or a vendor claim by separating what was measured from what was projected, and reply to a request for a twin whose output nobody could act on.
5. **Construct** a complete value case – decision, metric, baseline, pattern, fidelity requirement, cost and payback – for a small cyber-physical system, and state the one test that could falsify it before any code is written.

Every section that follows serves at least one of those five, and the exercises at the end say which.

First Meeting

Here is the situation this chapter is written for.

You are a software engineer. Someone senior has decided that your organisation needs a DT. There is a kick-off meeting. Around the table are a modelling expert who talks about solvers and boundary conditions, a business sponsor who talks about uptime and margin, and a customer who talks about the thing they own and worry about. You are there because someone has to build it.

The failure mode of that meeting is not technical. It is that everyone agrees enthusiastically while meaning four different things, and nobody says out loud which decision the twin is supposed to improve. Six months later the project has a beautiful three-dimensional model, a live data feed, a dashboard, and no answer to the question "what did we do differently because of it?"

This is not a hypothetical failure. Writing about supply-chain twins, one practitioner warns that new technology reads as an expensive gamble unless whoever is buying it keeps their hopes for the payoff, and for how soon it arrives, realistic [2]. A set of pilot reports from European manufacturing points the same way from the other side: the company that describes its pilot as a success says it had worked out, before starting, why it wanted the twin and what it would be used for [3]. A brief from an industry consortium adds the part that is hardest to hear in a kick-off: the process around the twin has to change for the benefits to arrive, because a twin bolted onto an unchanged process does not deliver them [4].

So the first skill this book teaches is not modelling and not architecture. It is the ability to sit in that meeting and ask, without embarrassment: **which decision gets better, who makes it, and how will we know?**

Where the idea came from, in one paragraph

You will hear the origin story in meetings, so it is worth having straight. The concept traces to a 2002 presentation to industry, given at the University of Michigan, in which Michael Grieves proposed a real space, a virtual space, and a data link between them for product lifecycle management; the early courses called it the Mirrored Spaces Model [5]. The name "digital twin" attached later, and the 2010 technology roadmap of the National Aeronautics and Space Administration (NASA) gave the first detailed definition, in the aerospace context [6]. The folklore version – that NASA's Apollo programme built two identical vehicles, flew one and kept the other on the ground to rehearse fixes – is real enough as history and is

routinely cited as the ancestor of the idea [7]. The defence-side literature makes the same point without the twin vocabulary: Apollo leaned on simulation throughout, to keep the risk down for the astronauts and for the people running mission control [8].

Note what the Apollo story is really about. It is not about fidelity for its own sake. It is about moving a decision – "will this fix work?" – from a place where being wrong is fatal to a place where being wrong is cheap. Nearly every DT value argument in this chapter is a variant of that one move.

Market Forecast

You will be shown a market forecast. Here is how to read one without either swallowing it or sneering at it.

The published figures are large and mutually inconsistent, and the four that reach this corpus through academic surveys are set side by side in Table 1 (figures in US dollars (USD)). Each row is a paper quoting an analyst house; none of the four papers measured a market.

Table 1: Four digital-twin market forecasts, as relayed by papers in this corpus. Every figure is an analyst projection quoted second-hand.

Quoted in	Analyst house	From	To	Growth quoted
A 2020 state-of-the-art survey [9]	Grand View Research	USD 2.26 billion (2017)	USD 27.06 billion (2025)	"approximately tenfold"
A 2022 enabling-technologies survey [10]	not named in the survey	USD 3.2 billion (2020)	USD 48.27 billion (2026)	58% per year
A 2022 modelling review [11]	Markets and Markets	USD 3.1 billion (2020)	USD 48.2 billion (2026)	not quoted
A 2025 twinning-systems paper [12]	Fortune Business Insights	USD 8.6 billion (2022)	USD 137.7 billion (2030)	42.6% per year

Read the table down the "To" column. Two forecasts for the same year, 2026, agree with each other to within a rounding error, which is less reassuring than it looks: they are the same analyst estimate reaching two papers by different routes. Set the 2025 and 2030 figures beside them and the implied size of the same market in any one year moves by more than a factor of two depending on who counted. Four observations follow, in order of usefulness to you:

1. **These are forecasts, not measurements.** Cite them as evidence of expectation, not of size. The same survey that relays the 58% growth figure also notes that Gartner named DTs a top-ten strategic technology trend for 2019 [9], which is evidence of attention, not of revenue.

2. **The disagreement is the information.** When "digital twin" can mean a dashboard or a coupled multi-physics simulation, any total is a total over an ill-defined set. That is why the definitional chapter exists.
3. **Adoption lags the forecasts, and the adoption numbers also disagree.** The 2022 survey relays an engineering-institution report putting industry-agnostic adoption at about 5% of enterprises [10], while the 2025 paper relays surveys in which only about 5% of companies say a twin plays *no* role in their digital-transformation plans [12]. Both can be true. "We have a strategy" and "we have a twin in production" are different claims, and holding both in mind is the correct posture.
4. **Sector reports beat global totals.** Oil and gas [13], health [14], smart cities [15] and net-zero manufacturing [16] each have their own literature with concrete application scenarios. A sector report tells you what people in *your* customer's industry are doing. A global total tells you what a research firm sold a report about.

Two further pieces of context are worth carrying into the meeting. Several years of digital-transformation progress are credited to the COVID-19 period alone [10], so part of the growth in these curves is a step change in attention rather than a smooth technology trend. And vendor material is part of the evidence base: an accessible overview of the field draws on what large vendors report doing with twins [17]. That is not disqualifying, but it means the loudest numbers come from parties with an interest in them.

What to say in the meeting. Not "the market will be USD 137 billion". Say: "forecasts are large and inconsistent, adoption is early, and our justification has to come from our own baseline, not from a growth curve." That sentence is both more defensible and more useful.

Key Question

Start from a claim that will look reductive and then earn its keep:

> **A DT is bought to improve a decision. If you cannot name the decision, there is no value case – only a budget.**

A decision has four parts you can write down in a meeting:

1. **The decision itself.** "How much reagent does vessel 7 get this shift?" "Do we take turbine 12 offline this week or next month?" "Do we accept this batch?"
2. **The decider, and the cadence.** A human once a quarter, a human every morning, or a control loop every 200 milliseconds. These are three different systems, and the cadence drives most of your architecture.
3. **What is wrong with the decision today.** It is made too late, or on incomplete information, or by a person who has to be physically present, or it is not made at all because nobody noticed there was a decision to make.
4. **What it costs to be wrong.** In currency (this chapter's worked examples are costed in euros, EUR), in downtime, in scrapped product, in a scrapped batch, in a life.

Point 4 is the one that gets skipped, and it is the one that determines whether the project is worth doing. A twin that improves a decision whose error cost is EUR 200 a year is a hobby.

The value metric

From those four parts you extract exactly one **value metric**: a measurable quantity, with a stated baseline and a stated way of measuring it, that the twin is supposed to move.

Insisting on *one* metric is the core of what practitioners call the **lean digital twin**: pick the single quantity the business most wants moved, attach a money value to moving it, build only what is needed to show the movement within weeks rather than years, and let that early return pay for the next, more capable increment [18]. That is a familiar shape to any engineer who has shipped a minimum viable product, and the analogy is the authors' own. The business viewpoint of an industrial reference architecture asks for the same thing in more formal clothes: a return, with costs and benefits tracked across the whole solution lifecycle – development, operation, evolution and maintenance, not the build alone [19].

Bad value metrics are abstract nouns: *efficiency*, *visibility*, *optimisation*, *digital transformation*. Good value metrics have units and a denominator, and Table 2 shows the difference on four requests you will actually hear.

Table 2: Requests as they arrive in a meeting, and the value metric each one hides.

A request as it arrives	The value metric behind it
"Improve maintenance"	Unplanned downtime hours per turbine per year
"Better dosing"	Batches scrapped because of undetected dosing faults per year
"More visibility"	Hours between a fault occurring and an engineer being told
"Optimise the line"	Scrapped units per 10,000 produced

The right-hand column has a property the left does not: someone can go and measure it *today*, before you build anything, and get the **baseline**. A value claim with no measured baseline is subtraction with one operand missing.

Three quantities carry the rest of this chapter, so pin them now. Write the value metric as M, its measured baseline as M_0, the value the twin is expected to reach as M_1, and the cost of one unit of the metric as c. The **annual benefit** is then

$$B = (M_0 - M_1)\,c \tag{1}$$

and the **payback period**, in months, is the build cost plus one year of running cost divided by that benefit:

$$P = 12\,\frac{C_{\text{build}} + C_{\text{run}}}{B} \tag{2}$$

Both are arithmetic a sponsor can check on a napkin, which is the point. Equation 1 is where most of the argument happens, because M_1 is a forecast and c is a judgement, and Equation 2 is where the argument is tested, because C_{build} and C_{run} are things you will be held to.

Decision latency: what twins usually actually sell

Look across the deployments in the Application Areas section and one quantity keeps reappearing: **decision latency**, the time between the PT entering some state and someone, or something, acting on it. Occasionally a twin opens up a decision nobody could have made before, because it offers a service that did not exist otherwise. Far more often it makes an already-possible decision happen *hours or weeks earlier*, or *without a human standing next to the equipment*.

Decision latency splits into two gaps, and a DT does not touch both of them equally. Figure 1 separates them.

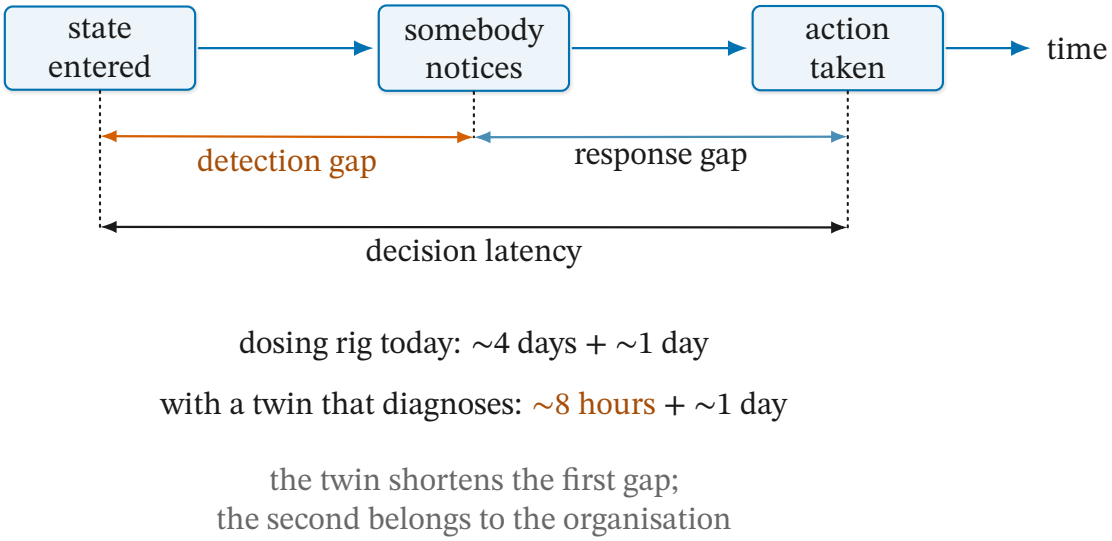


Figure 1: Decision latency, split into the detection gap a twin shortens and the response gap it does not.

Write the two as G_{d}, the **detection gap** from the state being entered to somebody noticing, and G_{r}, the **response gap** from noticing to acting. Then

$$L = G_{\mathrm{d}} + G_{\mathrm{r}} \tag{3}$$

Equation 3 looks too small to write down, and its value is in what it separates. The detection gap is an engineering problem and yours to solve. The response gap is an organisational one – how fast the lab, the crew or the maintenance schedule can act once told – and no amount of model fidelity shortens it. This is the concrete form of the consortium brief's warning that the process has to change for the benefit to arrive [4]: a twin that cuts G_{d} from four days to eight hours has done nothing for a plant whose G_{r} is a fixed weekly maintenance round. It is why the *predict* pattern in the next section carries a warning, and why "nobody can act on the output" appears below among the reasons to stop.

That split is worth drawing in the kick-off meeting, because it reframes the project from "build a virtual replica" – unbounded, expensive, un-finishable – to "cut the time from event to action from four days to four hours" – bounded, estimable, testable. The lean-twin literature's insistence on a benefit that can be verified quickly [18] is the same reframing from the finance side.

Three examples, with the support fading

The method above is small enough to run in an afternoon. Here it is run three times: once in full on a small dosing rig, once on a larger system with one step left to you, and once posed as a problem for the exercises.

Example 1, worked in full: a reagent dosing rig Picture a pilot plant with twelve reactor vessels. Each vessel has an inline concentration sensor and a small dosing pump fed from a shared reagent tank, all wired to one controller that stores every reading in a time-series database and exposes a REST (Representational State Transfer) API. Nothing about it is exotic; a brewery, a water-treatment works or a fish farm has the same shape. For the value case, one fact about it matters, and Figure 2 shows it.

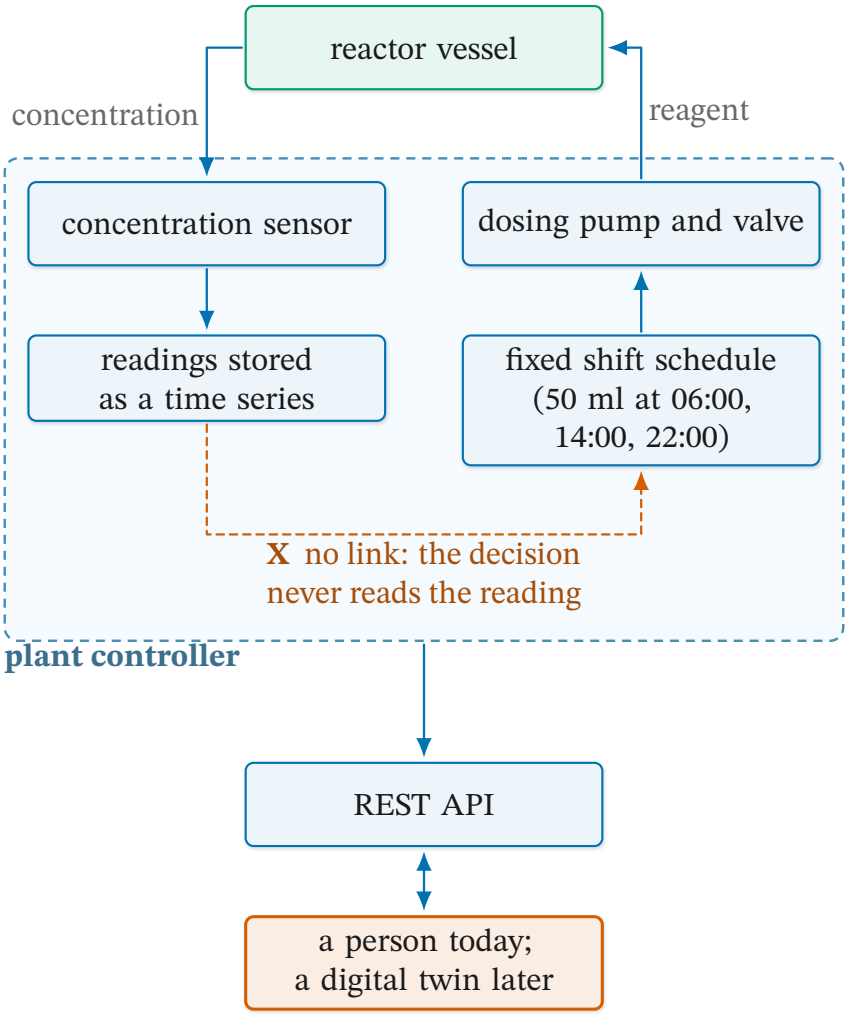


Figure 2: The dosing rig as it exists today. Sensing works and actuation works; the dosing decision never reads the measurement. The missing link, marked X, is what this value case is about.

Step 1 – the decision. Dosing is on a fixed shift schedule: 50 ml at 06:00, 14:00 and 22:00, whatever the sensor says. The decision "how much reagent, and when" is made once, by whoever set the timer, and then not made again. The sensor readings are stored and never consulted by anything that acts. So the decision that exists today is not "how much reagent" but "has dosing stopped working": a blocked line, an empty tank, a valve that no longer

opens. Today a person makes that decision when the end-of-batch assay comes back out of specification.

Step 2 – the decider and the cadence. A shift technician, on their rounds, roughly daily. The reasoning for writing this down: it fixes G_{r}. Once told, the technician can act within about a day, and no software changes that.

Step 3 – what is wrong with it today. The decision is made late. A dosing fault shows in the assay after about four days, so G_{d} is about four days and L is about five. The rig's own log confirms it: the stored concentration series shows flat traces for days before each recorded intervention.

Step 4 – what being wrong costs. This is where the obvious metric fails and the real one appears. The obvious metric is reagent. Twelve vessels at three doses of 50 ml a day is $12 \times 0.05 \times 3 \times 365 = 657$ litres a year, which at the reagent's bulk price is a few hundred euros. Nobody funds a project to save that. The real cost is that each vessel runs a six-day batch, and a batch that went undosed for more than about two days is scrapped and restarted. Last year's log records ten dosing faults and, counting the batches scrapped because of each, 37 batches lost. The plant costs a batch – materials plus vessel time – at about EUR 800. So the metric is *batches scrapped because of undetected dosing faults per year*, with $M_0 = 37$ and $c = 800$, and the baseline cost of the problem is $M_0\, c = 37 \times 800 = 29{,}600$ euros a year. The reasoning for choosing this metric over reagent: it is the quantity whose reduction the sponsor would recognise as the point.

Step 5 – the target. A twin that compares each dose's expected effect on the concentration with the recorded effect would see a fault at the first missed dose, within one dosing cycle of about eight hours. With G_{d} at eight hours and G_{r} at a day, the batch survives. Not every fault is visible this way – a sensor that fails silently is the obvious hole – so set the target conservatively at $M_1 = 4$. Then, by Equation 1, $B = (37 - 4) \times 800 = 26{,}400$ euros a year. The reasoning for the conservative target: a value case that survives its own pessimism is one you can defend in the meeting.

Step 6 – the pattern and the fidelity it needs. The twin has to notice that a dose did *not* do what a dose does. It does not need to forecast the concentration a week ahead, and it does not need to decide the dose. In the vocabulary of the next section that is *diagnose*, not *predict* and not *control*, and the model it requires has to get the direction and rough size of a dose's effect right – nothing more. The reasoning: fidelity is being derived from the metric, not chosen for its own sake.

Step 7 – cost and payback. Three engineer-weeks to build the comparison, alerting and a small calibration harness, plus half a week of a modeller's time for the dose model, at the plant's internal rate: $C_{\mathrm{build}} = 14{,}000$. Running cost a year: hosting at a few hundred euros, plus half a day a month of upkeep as vessels, tubing and sensors are swapped: $C_{\mathrm{run}} = 5{,}400$. By Equation 2, $P = 12 \times \frac{14{,}000 + 5{,}400}{26{,}400} = 8.8$ months.

Step 8 – the test that could kill it. Before writing any code, spend one afternoon querying the stored concentration history for the ten logged faults. If the traces do *not* go flat at the first missed dose – if the sensor noise swamps a 50 ml step, or the faults were of a kind the sensor cannot see – then M_1 is not 4, the eight-hour detection gap is fiction, and the case collapses. That afternoon costs nothing and can falsify the whole argument. Do it first.

Example 2, one step left to you: a pump fleet A water utility runs forty similar pumps across a city. Unplanned outages are logged: last year they totalled 1,200 pump-hours, so $M_0 = 1{,}200$ with the metric *unplanned outage hours per year across the fleet*. Each outage hour costs the utility about EUR 250 in standby crew and overflow penalties, so $c = 250$. The decision is "which pump do we visit today", made every morning by a maintenance planner, and today it is made from whichever pump complained loudest overnight.

A twin that flags a pump whose vibration and power draw drift from what its model expects – *diagnose* again – would let the planner visit before the failure. Engineering judgement, checked against the log, says about 40% of outage hours were preceded by a detectable drift, so set $M_1 = 720$. Then $B = (1{,}200 - 720) \times 250 = 120{,}000$ euros a year. Integration with forty pumps of three vintages, a vibration model per vintage and a year of running cost come to $C_{\text{build}} = 90{,}000$ and $C_{\text{run}} = 30{,}000$.

Your step: compute P from Equation 2, and then answer the question the payback does not: what is G_{r} here, and what would make the whole case fail even if the payback looks acceptable? A check appears in the exercise solutions.

Example 3, posed as a problem: a municipal park A parks department irrigates a large public park from a borehole on a fixed schedule. Water is metered and paid for. Turf is replaced when it dies. Complaints arrive by e-mail. The head of department has asked for "a digital twin of the park". Building this value case in full – decision, metric, baseline, target, pattern, fidelity, cost, payback, and the test that could kill it – is the hardest exercise at the end of this chapter, and it is deliberately not worked here.

When the right answer is no

The method has a feature its enthusiasts underplay: run honestly, it often returns "don't". Eight conditions, any one of which should stop the project or shrink it to a rung that pays:

1. **No decision can be named.** The request is for visibility, insight or transformation, and nobody can say what would be done differently.
2. **The error cost is trivial.** The metric is real but c is small enough that B never covers C_{run}.
3. **Nobody can act on the output.** G_{r} is fixed by a schedule, a contract or a staffing level, and shortening G_{d} changes nothing.
4. **The baseline cannot be measured.** No log, no history, no way to get M_0 without first building the thing that was supposed to justify itself.
5. **The falsifying test already failed.** The stored history does not show the signal the twin would need.
6. **A cheaper rung already delivers the benefit.** A threshold alarm on the existing sensor gets most of B at a fraction of C_{build}.
7. **The model cannot be kept valid.** The PT changes faster than anyone will recalibrate; the next section on cost says why this line is the one most often forgotten.
8. **The organisation has not agreed to change its process.** The brief on infrastructure twins is blunt that the twin on its own buys you nothing; somebody still has to be willing to work differently [4], and the pilot reports single out a clearly stated purpose as what separated the pilots that worked [3].

None of these is a reason to be against twins. They are the conditions under which the *lean*

literature's own logic – fund the increment that verifies quickly [18] – says there is no such increment yet.

Six Value Patterns

Across the deployment reports in this corpus, the *shapes* of value repeat. This book uses six names for them, and will use them consistently from here to the last chapter. They form a rough ladder: each rung typically needs everything below it, and each rung costs more than the one below. Figure 3 is the whole section on one page; the subsections that follow are its rungs, in order.

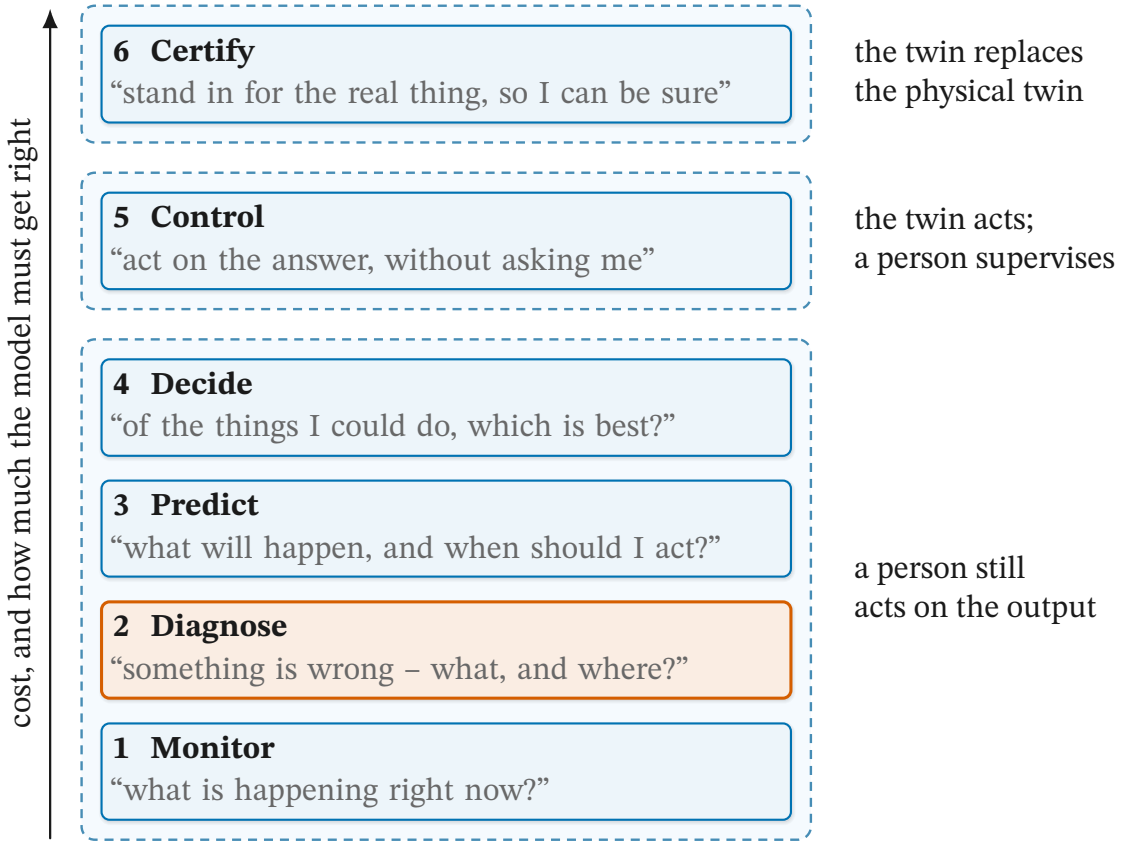


Figure 3: The six value patterns as a ladder. It is a vocabulary, not a maturity model: stop at the rung that pays for itself. The dashed groups mark who acts on the twin's output.

The ladder is a cost ordering, not a quality ordering. Rung 2 delivered is worth more than rung 4 proposed.

Pattern 1 – Monitor

"*What is happening right now, and what happened last Tuesday?*"

The twin ingests live measurements, stores them with context, and presents current and historical state. It may fill in quantities that are not directly measured: a **virtual sensor** is a model-derived estimate of a quantity with no physical sensor of its own – a weld's stress, say, that you cannot instrument directly – computed from the model's internal state [20]. A review of the field's evolution puts monitoring first historically too: twins began as

monitoring systems and were later extended towards control, optimisation and autonomy [21].

- **Buys you:** lower decision latency, remote access, an audit trail.
- **Costs you:** connectivity, data engineering, and the unglamorous work of keeping timestamps and units correct.
- **Honest warning:** many systems sold as DTs stop here.

Pattern 2 – Diagnose

"*Something is wrong. What, and where?*"

The twin compares what the model says *should* be happening with what the sensors say *is* happening. The difference is the **residual**:

$$r(t) = m_{\text{obs}}(t) - m_{\text{model}}(t) \tag{4}$$

where $m_{\text{obs}}(t)$ is the measured value and $m_{\text{model}}(t)$ the model's expectation at the same instant. A large, structured residual localises a fault; the simplest diagnostic rule flags one when $|r(t)| > \theta$ for longer than the model's own noise would explain, with the threshold θ set from the stored history. Choosing θ well is later work; what matters here is that Equation 4 is the entire mechanism, and it needs a model that gets the *direction and rough size* of a change right, not one accurate to three decimal places.

Figure 4 shows the rule at work on the dosing rig from the previous section. After each dose the model expects a step in concentration followed by a slow decline; the sensor reports a flat line. Three doses, three missing steps, three large residuals – the reagent is not reaching the vessel, and the fault is visible one dosing cycle after it starts rather than when the assay comes back.

Floating offshore wind's diagnostic twins run the same pattern: the twin watches asset condition for anomalies on equipment that costs a lot, and takes a while, to reach in person [22], and behind fault-diagnosis twins for rotating equipment more generally [23]. It is also the level a systems-of-systems maturity ladder for asset twins calls *adaptive*: the twin changes as the physical reality changes, within an acceptable uncertainty, and is enhanced with diagnostic capability [24].

- **Buys you:** faults caught in hours instead of days; fewer truck rolls to find out nothing is wrong.
- **Costs you:** a model good enough that a residual means something, plus the calibration work of a later chapter.
- **Note for the meeting:** diagnosis is the cheapest rung with real engineering value, and it is very often the right first increment.

Pattern 3 – Predict

"*What will happen, and when should I act?*"

The twin runs the model forward from the current estimated state to forecast future condition: remaining useful life, time to failure, time until a vessel's concentration drifts out

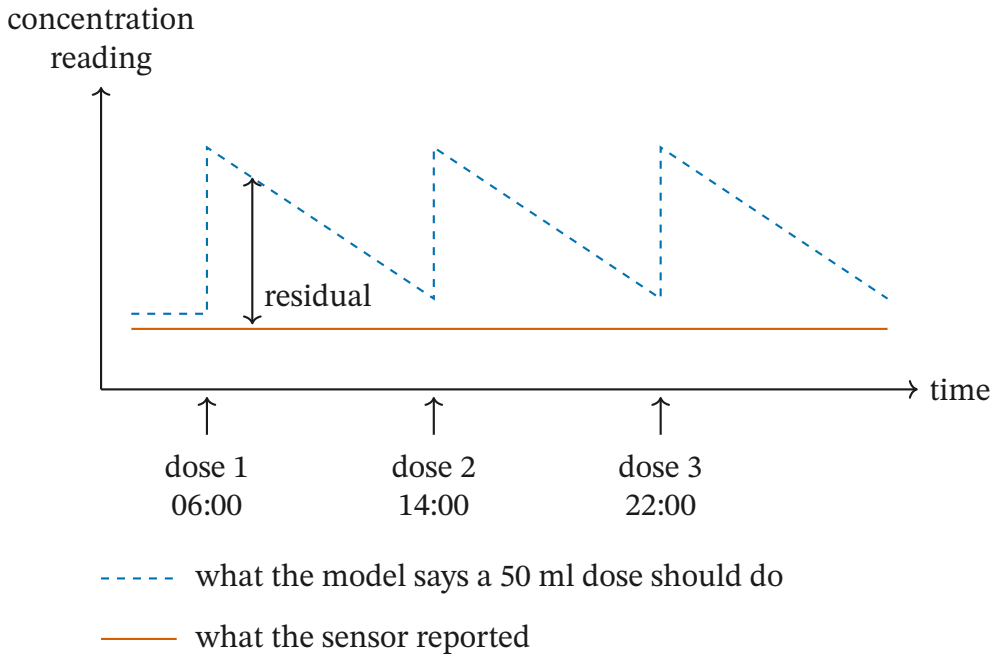


Figure 4: Why *diagnose* pays on the dosing rig. Each gap between the two traces after a dose is a residual, and the fault is visible within about eight hours rather than four days.

of specification. This is by far the most-reported family in the literature: predictive maintenance built on DTs has enough of its own literature to have been reviewed more than once [25], [26], and the maturity ladder above names this level *intelligent* – predictive and prescriptive capability layered on the adaptive twin [24].

- **Buys you:** maintenance scheduled by condition instead of by calendar; interventions before the failure rather than after.
- **Costs you:** everything in *diagnose*, plus a model that stays accurate over a forecast horizon, plus a serious answer to "how wrong might this be?" Uncertainty quantification stops being optional here [27].
- **Honest warning:** a prediction nobody acts on has zero value. If the maintenance crew's schedule is fixed six months ahead, a two-week-ahead prediction changes nothing. Check G_{r} before you sell a shorter G_{d}.

Pattern 4 – Decide

"*Of these five things I could do, which is best?*"

The twin runs many alternative futures and compares them. The literature calls this **what-if simulation**: a data- and computation-intensive approach that evaluates how the PT would behave under different scenarios and hypotheses, so that when something unwanted shows up an operator – or the twin itself – can simulate the candidate interventions, compare the outcomes, and choose between them on better information [28]. The output is a recommendation; a person still chooses.

A **fleet** is a set of similar PTs, each with its own DT, managed together, and the fleet level is where large operators claim their money back. A reported analysis of a 125-turbine fleet costs one component's early warning against the whole fleet rather than the single machine,

and that reframing is what turns cost control from a per-unit exercise into a plant-wide one – weighed against revenue, across the fleet [29].

- **Buys you:** better decisions, and decisions a human could not have enumerated by hand.
- **Costs you:** simulation you can run *fast* and *many times*. This is the first rung where compute cost becomes a design constraint rather than a line item: a what-if simulation is only useful if it runs faster than real time, and a documented incubator twin answers a power-outage plan by running three whole simulations, one per candidate controller setting, while the outage is still under way [28].

Pattern 5 – Control

"Act on the answer, without asking me."

The twin closes the loop: it changes the PT's behaviour directly, by reconfiguring it or by issuing set-points, and a person supervises rather than decides. The chapter on advanced services treats effecting change in the PT through reconfiguration as the natural end of what-if analysis [30], and the maturity ladder calls this level *autonomous*: the twin operates in automated closed-loop mode on the asset's own data, with autonomous functions built gradually on top of the automated ones [24].

The dosing rig's own closed loop – reading the concentration and setting the dose – is the obvious thing to build, and it is worth noticing that the value case in the previous section did *not* ask for it. Closing the loop would save a little reagent and would not shorten G_{d} for the faults that scrap batches. That is the ladder doing its job.

- **Buys you:** a decision that happens at the cadence of the PT rather than of a person – the 200-millisecond loop from the Key Question section.
- **Costs you:** everything below, plus the safety argument. The same systems-of-systems chapter that names the autonomous level also flags it as the contested one, and says it currently only fits cyber-physical systems that already have some autonomy built in [24]. If the twin is wrong, the PT does the wrong thing, unattended.

Pattern 6 – Certify

"Can I be confident before I touch the real thing?"

The twin substitutes for the PT when using the PT is expensive, slow, dangerous, or not yet built. The sub-case that matters most to you as a software engineer is **testing the software that controls the PT**. A reproducible exemplar built around a small robot vehicle uses a twin prototype precisely so that developers can swap a hardware-in-the-loop rig for a software-in-the-loop one and keep working on the same embedded software in parallel, without the special hardware environment that embedded testing usually demands [31]. Teams interviewed about continuous integration on cyber-physical systems report doing exactly this in their production pipelines – in one case relying on third-party simulators supplied by the customer because the hardware-in-the-loop rig lives in a clean room the pipeline cannot reach [32].

The second sub-case is **evidence**. Defence-side digital engineering is defined around an authoritative source of system data and models across the lifecycle, so that analysis done

on the models counts as evidence about the system [8]. In a regulated domain that is what "certify" means: the twin's output is admissible.

- **Buys you:** a test environment, a rehearsal space, and – in regulated domains – evidence.
- **Costs you:** credibility. If decisions rest on the twin's output, you must be able to say why it should be believed. That is what most projects underestimate.

Major Cost Contributors

Twins are not free, but the return can be large – that is the industrial sales pitch in one line, and it is a fair summary of the pitch as one European project's own material puts it [33]. Your job is to know which half of that sentence you are being sold. The reference-architecture view says costs run across development, operation, evolution and maintenance [19]; the six lines below follow that lifecycle from inception to disposal, roughly ordered by how often each is the surprise.

1. Getting the data out, and keeping it correct. Connectivity, protocol translation, timestamp discipline, units, and the context that says *which* vessel a concentration reading belongs to. This is your home turf and it is consistently underestimated. In the dosing rig's value case it is most of C_{build}, and it is the line the pilot reports itemise most concretely – one records the operating cost of radio-frequency identification (RFID) tags per can, and the licence terms for running its simulation software in the cloud [3].

2. Building and – worse – maintaining the model. The first version of a model is a project. Keeping it valid as the PT wears, gets repaired, or has a component swapped is a *permanent* cost. The twin has to track both continuous change, such as wear and tear, and discontinuous change, such as a component replaced after a failure [34]. The pilot reports reach the same place from the field: their final step, keeping the twin synchronised with the physical world and its knowledge managed, is the one they say will dominate the twin's life after the pilot ends [3]. This is the C_{run} line, and it is the cost most business cases omit entirely.

3. People who can do this. A systematic review of process-industry adoption names the lack of specialists and expertise as a barrier and workforce upskilling as the corresponding enabler [35]. A perspective on scaling twins puts the structural version of the same point: twins today are labour-intensive, highly specialised, *artisanal* products, and the barrier to wider adoption is that the methods do not transfer across disciplines [36].

4. Compute and licences. Real, and the least alarming of the six. The same process-industry review lists lower and decreasing hardware costs among the enablers rather than the barriers [35], and a construction-engineering study reports that running a twin atop managed cloud services made it both cheaper to build and cheaper to keep running [37]. Simulator reuse – third-party or customer-supplied simulators instead of writing your own – is a documented cost-reduction tactic [32].

5. Bespoke integration, over and over. The structural problem of the field: most twins are hand-built one-offs, and the perspective above argues that scaling them from artisanal to industrial requires the barriers to adoption to come down [36]. Every hour you spend re-inventing an interface that a standard already defines is cost with no value attached. Standardised reference architectures such as ISO 23247, the International Organization for

Standardization's reference architecture for twins in manufacturing, exist to stop that – and an analysis of documented architectures finds them only partly aligned with it, which is itself a measure of how much re-invention is going on [38].

6. Disposal. A twin outlives neither its PT nor its sponsor. When the rig is dismantled or the turbine retired, someone has to decide what happens to years of stored history, the calibrated models, and the interfaces other systems came to depend on. The pilot reports' "cradle to grave" framing [3] and the lifecycle view of evolution [34] both put decommissioning inside the twin's life, not after it. It is rarely costed at the start, and it is cheap to cost.

The shape of the estimate

For a first-increment twin, a serviceable rule for the kick-off meeting is this: **expect integration and data work to be the largest single line, expect model maintenance to be the largest line you forgot, and expect compute to be smaller than everyone fears.** The dosing rig's estimate in the Key Question section has exactly that shape – integration dominates C_{build}, upkeep dominates C_{run}, hosting is a rounding error. Then go and measure, because a rule of thumb is not an estimate.

Application Areas

The point of this section is not the technology. It is to show you four *different value arguments*, each with a different reason the money is there. Read each one asking: what is the decision, and what does being wrong cost?

Aerospace and defence – the cost of being wrong is enormous

Two US aerospace bodies – the American Institute of Aeronautics and Astronautics (AIAA) and the Aerospace Industries Association (AIA) – jointly published a position paper laying out the industry's rationale for accelerating DT adoption, and it is candid that the significant work under way is often siloed and uncoordinated within each organisation's own view of the system [39]. The defence side frames the same territory as *digital engineering*: an integrated approach using authoritative sources of system data and models across the lifecycle, with a lineage traced to Apollo-era simulation for risk reduction [8].

- **Decision:** whether this airframe or engine can safely fly another number of cycles; whether a design change is acceptable.
- **Why it pays:** the cost of being wrong includes lives and airframes. Structural-health work in this family uses twin approaches for life prediction under unpredictable operational conditions [40].
- **What it demands of you:** credibility and traceability above all. Not speed. This is *certify* territory.

Rotating machinery and offshore assets – the cost of *going there* is enormous

Offshore wind and offshore platforms have a distinctive economics: the asset is remote, visiting it is expensive and weather-dependent, and a failure found early is worth far more than the same failure found late. That is why the sector is a heavy adopter of condition-based and predictive maintenance with twins as the enabler [22]. North Sea offshore

platforms have had their fatigue tracked industrially with a twin approach, built up in five levels running from screening up to tracking accumulated fatigue [40]. A vendor's digital wind farm, giving every turbine its own twin to sharpen maintenance strategy, turns up in practically every list of standard applications [41].

- **Decision:** send a vessel and a crew, or don't.
- **Why it pays:** each avoided unnecessary trip, and each avoided catastrophic-but-preventable failure, is directly costed. This is *diagnose* and *predict* territory, and G_r – the sailing – is long, which is exactly why shortening G_d is worth so much.
- **What it demands of you:** intermittent, lossy connectivity; edge processing; long-horizon data.

Process and discrete manufacturing – the cost of being *slightly* wrong, multiplied

Here no single error is catastrophic, but tiny per-unit losses multiply across enormous volumes. The literature is correspondingly focused on the enablers and barriers to getting twins into the plant at all [35], and on where value shows up: per-component early warning and plant-wide cost control across a turbine fleet [29], reference architectures for predictive maintenance [42], and platform selection against a specific goal such as net-zero targets [16].

It is not only for giants. One case study follows a small-to-medium manufacturer of roll-to-roll printed labels that deliberately kept its technology spend small and its risk lower still, while still protecting the return [43] – the lean-increment argument of the Key Question section showing up in the field.

- **Decision:** run, adjust, or stop; accept or reject a batch.
- **Why it pays:** volume. A 0.3% scrap reduction is a real number when the denominator is millions.
- **What it demands of you:** integration with systems that predate you, and uptime.

Health, cities and infrastructure – the cost is societal, and the twin is contested

Health DTs have grown fast enough that the field already needs scoping reviews to keep track of itself [14], and it is organised enough to have published a position paper on the barriers slowing adoption and on what a shared ecosystem would need [44]. Smart-city work maps twin value onto four concrete municipal problems – running energy systems, keeping mobility sustainable, managing water resources, and responding to disasters – and onto evidence-based decision making for a community [15].

- **Decision:** a clinical or a policy one.
- **Why it pays:** outcomes, and cost avoidance at population scale.
- **What it demands of you:** governance, privacy, and an unusually high standard for explaining what the model does. This is the family where "the model said so" is least acceptable as an answer, and where *control* is least likely to be permitted.

The common thread

In all four, the twin is paid for because **a decision was being made too late, too blind, or too expensively** – and the twin moves it. None of the four is paid for because a virtual replica is interesting.

Summary

Back to the objectives this chapter opened with.

1. **Decision first.** Buy a twin to make one decision better. Name the decision, the decider, the cadence, and the cost of being wrong; compress them into one value metric with a measured baseline M_0; compute the benefit from Equation 1 and the payback from Equation 2. Verify a defined financial benefit quickly and let it fund the next increment [18].
2. **Decision latency is what twins usually sell**, and only half of it is yours: Equation 3 splits it into a detection gap engineering can shorten and a response gap the organisation owns [4].
3. **Six value patterns.** *Monitor, diagnose, predict, decide, control, certify*. Each rung typically needs the ones below and costs more. A person acts for the first four; at *control* the twin itself acts, and at *certify* it stands in for the PT altogether. *Diagnose* – built entirely around the residual from Equation 4 – is very often the right first increment.
4. **Cost structure.** Data integration is the biggest line, model maintenance is the biggest forgotten line [34], expertise is the most-reported barrier [35], compute is smaller than people fear, bespoke one-off integration is the field's structural cost problem [36], and disposal is the line nobody costs.
5. **Reading the evidence.** Market forecasts are large, inconsistent and analyst-sourced; adoption lags them; sector reports beat global totals. Four deployment families pay for twins for four different reasons, and all four reduce to moving a decision earlier, cheaper or safer. Eight conditions say *don't build it*.
6. **A complete value case.** For the dosing rig, the obvious metric (reagent) was worth a few hundred euros a year; the real metric (scrapped batches) was worth EUR 29,600; the right pattern was *diagnose*, not the flashier closed loop; the fidelity requirement fell out of the metric; payback was about nine months; and one afternoon of querying stored history can falsify the whole thing before you write any code.

The one sentence to carry forward: **fidelity is a requirement derived from a value metric, not a virtue pursued for its own sake.**

Exercises

Solutions or hints follow the exercises; each says which. The objective each exercise serves is given in brackets.

Exercise 1 (Objective 1, easy). For each request, write the value metric behind it with a unit, a denominator and a stated measurement method: (a) "improve building energy performance"; (b) "reduce risk in the substation"; (c) "make the production line smarter".

Exercise 2 (Objective 2, easy). Classify each system into one of the six patterns, say who acts on its output, and state the *next* rung up the ladder and what it would additionally require: (a) a dashboard showing live bearing temperature across forty pumps; (b) a system that flags a pump whose vibration deviates from its model; (c) a system that estimates each pump's remaining useful life; (d) a system that chooses which of forty pumps to service during a six-hour maintenance window; (e) a system that throttles a pump automatically when its model predicts cavitation.

Exercise 3 (Objective 1, medium). Finish Example 2. Compute P for the pump fleet from Equation 2, then name G_{r} for that decision and state the one organisational fact that would make the case fail even at that payback.

Exercise 4 (Objective 4, medium). A customer asks for a twin that predicts equipment failure two weeks ahead. Their maintenance schedule is fixed quarterly and cannot be changed within a quarter. Write the two-sentence reply you would give in the meeting, and state what you would propose instead.

Exercise 5 (Objective 4, medium). A vendor's slide says "the digital twin market will reach USD 137 billion by 2030". Using the Market Forecast section, write three questions you would ask before repeating the number, and say what each answer would change.

Exercise 6 (Objective 3, medium). Rank the six cost lines of the Major Cost Contributors section for (a) the dosing rig and (b) the pump fleet, and say for each which line you would expect the sponsor to have left out of their own estimate.

Exercise 7 (Objectives 1 and 5, hard). Do Example 3, the municipal park, in full: decision, metric, baseline, target, pattern, fidelity requirement, cost, payback, and the test that could kill it. Then write the single paragraph you would put at the top of the proposal for a reader who will read only that paragraph.

Exercise 8 (Objective 3, hard, open-ended). The cost section claims model *maintenance* is the most-forgotten cost. Take a software system you have personally maintained and estimate the ratio of its total maintenance cost to its original build cost. Then argue whether a physics model's ratio should be higher or lower, and why. There is no correct answer; there is a correct *form* of answer, which names the drivers of change on both sides [34].

Solutions and hints

Exercise 1 (solution). (a) "kWh per square metre per heating-degree-day, from the building meter, measured over one heating season." (b) "Unplanned outage minutes per year, from the Supervisory Control and Data Acquisition (SCADA) event log" – note that "risk" is not a metric until you say risk *of what*, measured *how*. (c) There is no answer, and that is the point: "smarter" names no decision. The correct response is the four questions of the Key Question section.

Exercise 2 (solution). (a) *monitor*, a person acts; next is *diagnose*, which additionally requires a model of expected temperature and a residual. (b) *diagnose*, a person acts; next is *predict*, which additionally requires a forecast horizon and an uncertainty estimate [27]. (c) *predict*, a person acts; next is *decide*, which additionally requires running many scenarios fast [28]. (d) *decide*, a person still chooses; next is *control*, which additionally requires a safety argument for acting unattended. (e) *control*, the twin acts and a person supervises.

Exercise 3 (solution). $P = 12 \times \frac{90{,}000+30{,}000}{120{,}000} = 12$ months. G_{r} is the planner's morning round plus the crew's travel – about a day. The fact that kills the case is a maintenance contract that fixes visits by calendar: then G_{d} shortens and nothing else moves, which is condition 3 of "When the right answer is no".

Exercise 4 (hint). Condition 3 of "When the right answer is no", and the warning under *predict*. A good reply names the mismatch without blaming the customer, and proposes the

rung below: *diagnose*, which is actionable within their existing quarterly cycle because it tells them what to put *on* the next quarter's list.

Exercise 5 (hint). Who measured it, and what did they count as a digital twin? What was the adoption figure in the same report? Which sector figure, rather than the global total, applies to this customer? Each answer moves the number from evidence of size to evidence of expectation.

Exercise 6 (hint). For the dosing rig, expect the sponsor to have costed compute and forgotten model upkeep; for the fleet, expect three pump vintages to make integration the surprise and the vibration models the forgotten line.

Exercise 7 (hint). There is no solution, but there is a test. If your paragraph contains a number without a source, it is not finished. Start from the water meter and the turf-replacement invoices: those are the two baselines that already exist.

Exercise 8 (open). Grade it on whether the answer names a driver of change on each side: requirements churn and dependency updates for the software; wear, repair, swapped components and PT reconfiguration for the physics model [34].

Further Reading

- *A full book on engineering cyber-physical twins*, the closest fit to this book's angle: the opening chapter of an edited volume [45], its chapter of worked case studies [46], and its chapter on advanced services and what-if analysis [30], all in [47].
- *Where the concept came from:* [5] for the 2002 origin, [6] for the naming history and the NASA definition, [7] for the Apollo lineage, and [48] for the question of whether twins are an evolution of modelling and simulation or something new.
- *Surveys with breadth:* [10] on enabling technologies and trends, [41] on tools, [9] on the business-model angle, [49] on the networking and Internet-of-Things side, [21] on the concept's evolution, and [12] on moving from twins to twinning systems.
- *Value and adoption specifically:* [18] on the lean approach this chapter's method is built on, [35] on enablers and barriers, [2] on being realistic about return on investment, [4] on why the process has to change too, [19] for the business viewpoint of a reference architecture, [3] and [33] on pilot experience, and [43] for a small-manufacturer case.
- *Autonomy and the levels people will quote at you:* [50] for the 5C architecture, [24] for the descriptive-to-autonomous maturity ladder, [51] for the model / shadow / twin cut by data flow, and [52] for the monitor-analyse-plan-execute over a shared knowledge base (MAPE-K) loop behind autonomic twins.
- *Sector deep-dives:* [13] for oil and gas, [14] and [44] for health, [15] for smart cities, [39] and [8] for aerospace and defence, [40] for offshore structural health, and [16] for net-zero manufacturing platforms.
- *Small, reproducible exemplars* of the kind the worked example imitates: [53] presents a greenhouse twin as a community exemplar, [54] covers lifecycle management on a comparable set-up, [55] is a field-scale agricultural twin on an open platform, [56] catalogues architecture patterns using agriculture cases, and [31] is a small-scale reproducible exemplar built around a robot vehicle.
- *Consulted but not drawn on above,* and worth knowing they exist: [38] on ISO 23247, [25], [26] and [42] on predictive maintenance, [20] on the executable twin and model order reduction, [22] and [23] on diagnostic twins, [57] on what a twin has to *know* to

be trusted, [29] on industrial applications, [27] and [11] on modelling and uncertainty, [17] as an accessible overview, [32] on continuous integration for cyber-physical systems, [37] on running a twin on managed cloud services, [36] on scaling, and [34] on evolution as the core of twin engineering. Further afield, on architecture and reference models: [58] on a cloud-based reference architecture for cyber-physical systems, [59] on architecting twins in general, and [60], [61] on a unifying reference model for cyber-physical twins. On engineering-design and manufacturing use: [62], [63] on developing and leveraging twins in engineering design, [64] on model-based engineering of multi-purpose manufacturing twins, [65] on a digital shadow for robot motion control, [66] on reduced-order models for predictive twins, and [67] on a cyber-physical maintenance case study. On lifecycle, testing and data: [68] alongside its companion entry above on lifecycle management, [69] on microservice-based multi-level testing, [70] on integrating past, present and future data into one model, and [71] on Asset Administration Shell communication. On security and health: [72] on twin cybersecurity in Industry 4.0, [73] as a second scoping review alongside the one cited above, and [74] on a molecular twin for cancer-outcome prediction. And [75] is the edited volume that [15]'s smart-cities chapter, cited above, appears in; and [76] applies pretrained language models to forecasting when-it-happens outcomes for twinned equipment.

References

[1] T. Bergs, S. Gierlings, T. Auerbach, A. Klink, D. Schraknepper, and T. Augspurger, "The concept of digital twin and digital shadow in manufacturing," *Procedia CIRP*, vol. 101, pp. 81–84, 2021, Accessed: Jan. 21, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2212827121006612

[2] R. Bhandal, "Conceptualising the Application of Digital Twins in Supply Chain Management: A Path Towards Supply Chain Resilience," in *Digital Twin: Fundamentals and Applications*, S. Sabri, K. Alexandridis, and N. Lee, Eds., Cham: Springer Nature Switzerland, 2024, pp. 173–189. doi: 10.1007/978-3-031-67778-6_8.

[3] "Case studies of digitalization for creating digital twins." Change2Twin project. Available: https://www.change2twin.eu/about/deliverables/

[4] T. Milligan, "Infrastructure Lifecycle: A Case for Change."

[5] M. Grieves and J. Vickers, "Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems," in *Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches*, F.-J. Kahlen, S. Flumerfelt, and A. Alves, Eds., Cham: Springer International Publishing, 2017, pp. 85–113. doi: 10.1007/978-3-319-38756-7_4.

[6] O. J. Pinon Fischer, S. Sabri, and Y. Chen, "Fundamentals of Digital Twins, Modeling Approaches, and Governance," in *Digital Twin: Fundamentals and Applications*, S. Sabri, K. Alexandridis, and N. Lee, Eds., Cham: Springer Nature Switzerland, 2024, pp. 13–29. doi: 10.1007/978-3-031-67778-6_2.

[7] F. Tao, M. Zhang, and A. Y. C. Nee, Eds., "Digital Twin Driven Smart Manufacturing," in *Digital Twin Driven Smart Manufacturing*, Academic Press, 2019, pp. i–iii. doi: 10.1016/B978-0-12-817630-6.00013-8.

[8] "Digital Engineering Effectiveness." May 2022. Accessed: Mar. 13, 2024. [Online]. Available: https://insights.sei.cmu.edu/library/digital-engineering-effectiveness/

[9] K. Y. H. Lim, P. Zheng, and C.-H. Chen, "A state-of-the-art survey of Digital Twin: Techniques, engineering product lifecycle management and business innovation perspectives," *Journal of Intelligent Manufacturing*, vol. 31, no. 6, pp. 1313–1337, Aug. 2020, doi: 10.1007/s10845-019-01512-w.

[10] S. Mihai *et al.*, "Digital Twins: A Survey on Enabling Technologies, Challenges, Trends and Future Prospects," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 4, pp. 2255–2291, 2022, doi: 10.1109/COMST.2022.3208773.

[11] A. Thelen *et al.*, "A comprehensive review of digital twin—part 2: Roles of uncertainty quantification and optimization, a battery digital twin, and perspectives," *Structural and Multidisciplinary Optimization*, vol. 66, no. 1, p. 1, Dec. 2022, doi: 10.1007/s00158-022-03410-x.

[12] G. Lugaresi and H. Vangheluwe, *From Digital Twins to Twinning Systems*. 2025.

[13] T. R. Wanasinghe *et al.*, "Digital Twin for the Oil and Gas Industry: Overview, Research Trends, Opportunities, and Challenges," *IEEE Access*, vol. 8, pp. 104175–104197, 2020, doi: 10.1109/ACCESS.2020.2998723.

[14] E. Katsoulakis *et al.*, "Digital twins for health: A scoping review," *npj Digital Medicine*, vol. 7, no. 1, pp. 1–11, Mar. 2024, doi: 10.1038/s41746-024-01073-0.

[15] D. McKee and D. Dokter, "DISCS: An Approach for Accelerating the Development of Digital Twins for Smart Cities," in *Digital Twin: Fundamentals and Applications*, S. Sabri, K. Alexandridis, and N. Lee, Eds., Cham: Springer Nature Switzerland, 2024, pp. 31–58. doi: 10.1007/978-3-031-67778-6_3.

[16] D. Parle, G. Sharma, N. Anand, N. Padgaonkar, D. Stoddart, and D. Malley, "A Comparative Analysis for Harnessing Digital Twin Platforms for Net-Zero Manufacturing," *IEEE Access*, vol. PP, Aug. 2024, doi: 10.1109/ACCESS.2024.3447475.

[17] R. Saracco, "Digital Twins: Bridging Physical Space and Cyberspace," *Computer*, vol. 52, no. 12, pp. 58–64, Dec. 2019, doi: 10.1109/MC.2019.2942803.

[18] P. van Schalkwyk and D. Isaacs, "Achieving Scale Through Composable and Lean Digital Twins," in *The Digital Twin*, N. Crespi, A. T. Drobot, and R. Minerva, Eds., Cham: Springer International Publishing, 2023, pp. 153–180. doi: 10.1007/978-3-031-21343-4_6.

[19] "The Industrial Internet Reference Architecture," *Industry IoT Consortium*. Accessed: Jan. 08, 2025. [Online]. Available: https://www.iiconsortium.org/iira/

[20] D. Hartmann and H. Van der Auweraer, *The Executable Digital Twin: Merging the digital and the physics worlds*. 2022.

[21] M.-E. Iliuţă, M.-A. Moisescu, E. Pop, A.-D. Ionita, S.-I. Caramihai, and T.-C. Mitulescu, "Digital Twin—A Review of the Evolution from Concept to Technology and Its Analytical Perspectives on Applications in Various Fields," *Applied Sciences*, vol. 14, no. 13, p. 5454, Jan. 2024, doi: 10.3390/app14135454.

[22] F. Stadtmann and A. Rasheed, "Diagnostic Digital Twin for Anomaly Detection in Floating Offshore Wind Energy." arXiv, Jun. 2024. doi: 10.48550/arXiv.2406.02775.

[23] H. Zhang *et al.*, "Digital-Triplet: A new three entities digital-twin paradigm for equipment fault diagnosis," *Journal of Intelligent Manufacturing*, Aug. 2024, doi: 10.1007/s10845-024-02471-7.

[24] E. Altamiranda, "A System of Systems Foundation for Digital Asset Lifecycle Management," in *Digital Twin: Fundamentals and Applications*, S. Sabri, K. Alexandridis, and N. Lee, Eds., Cham: Springer Nature Switzerland, 2024, pp. 59–87. doi: 10.1007/978-3-031-67778-6_4.

[25] R. van Dinter, B. Tekinerdogan, and C. Catal, "Predictive maintenance using digital twins: A systematic literature review," *Information and Software Technology*, vol. 151, p. 107008, Nov. 2022, doi: 10.1016/j.infsof.2022.107008.

[26] D. Zhong, Z. Xia, Y. Zhu, and J. Duan, "Overview of predictive maintenance based on digital twin technology," *Heliyon*, vol. 9, no. 4, p. e14534, Apr. 2023, doi: 10.1016/j.heliyon.2023.e14534.

[27] A. Thelen *et al.*, "A comprehensive review of digital twin — part 1: Modeling and twinning enabling technologies," *Structural and Multidisciplinary Optimization*, vol. 65, no. 12, p. 354, Nov. 2022, doi: 10.1007/s00158-022-03425-4.

[28] C. Gomes *et al.*, "Digital Twin Tutorial: The Incubator Case Study," in *Engineering Trustworthy Software Systems: 6th International School, SETSS 2024, Chongqing, China, April 14–21, 2024, Tutorial Lectures*, J. P. Bowen, C. Gomes, and Z. Liu, Eds., Singapore: Springer Nature, 2025, pp. 68–101. doi: 10.1007/978-981-96-4656-2_3.

[29] Y. Jiang, S. Yin, K. Li, H. Luo, and O. Kaynak, "Industrial applications of digital twins," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 379, no. 2207, p. 20200360, Aug. 2021, doi: 10.1098/rsta.2020.0360.

[30] M. Frasheri, T. Böttjer, P. G. Larsen, L. Esterle, and C. Gomes, "Advanced Digital Twin Services," in *The Engineering of Digital Twins*, J. Fitzgerald, C. Gomes, and P. G. Larsen, Eds., Cham: Springer International Publishing, 2024, pp. 209–222. doi: 10.1007/978-3-031-66719-0_10.

[31] A. Barbie and W. Hasselbring, "Toward Reproducibility of Digital Twin Research: Exemplified with the PiCar-X." arXiv, Aug. 2024. doi: 10.48550/arXiv.2408.13866.

[32] F. Zampetti, D. Tamburri, S. Panichella, A. Panichella, G. Canfora, and M. Di Penta, "Continuous Integration and Delivery Practices for Cyber-Physical Systems: An Interview-Based Study," *ACM Trans. Softw. Eng. Methodol.*, vol. 32, no. 3, pp. 73:1–73:44, Apr. 2023, doi: 10.1145/3571854.

[33] "Change2Twin," *Change2Twin Project*. Accessed: Nov. 29, 2024. [Online]. Available: https://www.change2twin.eu/about/download/

[34] T. Alskaif *et al.*, "Evolution at the Core of Digital Twin Engineering," Grand Rapids, USA: IEEE, Jul. 2025. doi: 10.5283/epub.77656.

[35] M. Perno, L. Hvam, and A. Haug, "Implementation of digital twins in the process industry: A systematic literature review of enablers and barriers," *Computers in Industry*, vol. 134, p. 103558, Jan. 2022, doi: 10.1016/j.compind.2021.103558.

[36] S. A. Niederer, M. S. Sacks, M. Girolami, and K. Willcox, "Scaling digital twins from the artisanal to the industrial," *Nature Computational Science*, vol. 1, no. 5, pp. 313–320, May 2021, doi: 10.1038/s43588-021-00072-5.

[37] P. Zech, C. Nardin, S. Ristov, M. Flora, and R. Breu, "Digital-Twins-as-a-Service in Construction Engineering," in *2024 IEEE 20th International Conference on Automation Science and Engineering (CASE)*, Aug. 2024, pp. 3004–3010. doi: 10.1109/CASE59546.2024.10711409.

[38] E. Ferko, A. Bucaioni, P. Pelliccione, and M. Behnam, "Standardisation in Digital Twin Architectures in Manufacturing," in *2023 IEEE 20th International Conference on Software Architecture (ICSA)*, Mar. 2023, pp. 70–81. doi: 10.1109/ICSA56044.2023.00015.

[39] "Digital Twin: Definition & Value – An AIAA and AIA Position Paper," *Aerospace Industries Association*. Accessed: Apr. 23, 2024. [Online]. Available: https://www.aia-aerospace.org/publications/digital-twin-definition-value-an-aiaa-and-aia-position-paper/

[40] H. Pezeshki, H. Adeli, D. Pavlou, and S. C. Siriwardane, "State of the art in structural health monitoring of offshore and marine structures," *Proceedings of the Institution of Civil Engineers - Maritime Engineering*, vol. 176, no. 2, pp. 89–108, Apr. 2023, doi: 10.1680/jmaen.2022.027.

[41] Q. Qi *et al.*, "Enabling technologies and tools for digital twin," *Journal of Manufacturing Systems*, vol. 58, pp. 3–21, Jan. 2021, doi: 10.1016/j.jmsy.2019.10.001.

[42] R. van Dinter, B. Tekinerdogan, and C. Catal, "Reference architecture for digital twin-based predictive maintenance systems," *Computers & Industrial Engineering*, vol. 177, p. 109099, Mar. 2023, doi: 10.1016/j.cie.2023.109099.

[43] P. Singh, N. Nidhi, J. Karpavice, M. Beliatis, and M. Presser, *Digital Dataspace and Business Ecosystem Growth for Industrial Roll-to-Roll Label Printing Manufacturing: A Case Study*. 2023.

[44] M. Viceconti, M. De Vos, S. Mellone, and L. Geris, "Position Paper From the Digital Twins in Healthcare to the Virtual Human Twin: A Moon-Shot Project for Digital Health Research," *IEEE Journal of Biomedical and Health Informatics*, vol. 28, no. 1, pp. 491–501, Jan. 2024, doi: 10.1109/JBHI.2023.3323688.

[45] P. G. Larsen, J. Fitzgerald, and C. Gomes, "Engineering Digital Twins for Cyber-Physical Systems," in *The Engineering of Digital Twins*, J. Fitzgerald, C. Gomes, and P. G. Larsen, Eds., Cham: Springer International Publishing, 2024, pp. 3–17. doi: 10.1007/978-3-031-66719-0_1.

[46] B. J. Oakes *et al.*, "Case Studies in Digital Twins," in *The Engineering of Digital Twins*, J. Fitzgerald, C. Gomes, and P. G. Larsen, Eds., Cham: Springer International Publishing, 2024, pp. 257–310. doi: 10.1007/978-3-031-66719-0_12.

[47] J. Fitzgerald, C. Gomes, and P. G. Larsen, Eds., *The Engineering of Digital Twins*. Cham: Springer International Publishing, 2024. doi: 10.1007/978-3-031-66719-0.

[48] Z. Ali, R. Biglari, J. Denil, J. Mertens, M. Poursoltan, and M. K. Traoré, "From modeling and simulation to Digital Twin: Evolution or revolution?" *SIMULATION*, vol. 100, no. 7, pp. 751–769, Jul. 2024, doi: 10.1177/00375497241234680.

[49] A. Hakiri, A. Gokhale, S. B. Yahia, and N. Mellouli, "A comprehensive survey on digital twin for future networks and emerging Internet of Things industry," *Computer Networks*, vol. 244, p. 110350, May 2024, doi: 10.1016/j.comnet.2024.110350.

[50] P. G. Larsen *et al.*, "HUBCAP: A Novel Collaborative Approach to Model-Based Design of Cyber-Physical Systems," in *Simulation and Modeling Methodologies, Technologies and Applications*, M. S. Obaidat, T. Oren, and F. D. Rango, Eds., Cham: Springer International Publishing, 2022, pp. 90–110. doi: 10.1007/978-3-030-84811-8_5.

[51] W. Kritzinger, M. Karner, G. Traar, J. Henjes, and W. Sihn, "Digital Twin in manufacturing: A categorical literature review and classification," *Ifac-PapersOnline*, vol. 51, no. 11, pp. 1016–1022, 2018, Accessed: Dec. 11, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896318316021

[52] L. F. Rivera, M. Jiménez, N. M. Villegas, G. Tamura, and H. A. Müller, "The forging of autonomic and cooperating digital twins," *IEEE Internet Computing*, vol. 26, no. 5, pp. 41–49, 2021, Accessed: Jan. 12, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9325563/

[53] E. Kamburjan *et al.*, "GreenhouseDT: An Exemplar for Digital Twins," in *Proceedings of the 19th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, in SEAMS '24. New York, NY, USA: Association for Computing Machinery, Jun. 2024, pp. 175–181. doi: 10.1145/3643915.3644108.

[54] E. Kamburjan, N. Bencomo, S. L. Tapia Tarifa, and E. B. Johnsen, "Declarative Lifecycle Management in Digital Twins," in *Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems*, Linz Austria: ACM, Sep. 2024, pp. 353–363. doi: 10.1145/3652620.3688248.

[55] S. Infante *et al.*, "Integrating FMI and ML/AI models on the open-source digital twin framework OpenTwins," *Software Practice and Experience*, Mar. 2024, doi: 10.1002/spe.3322.

[56] B. Tekinerdogan and C. Verdouw, "Systems Architecture Design Pattern Catalog for Developing Digital Twins," *Sensors*, vol. 20, no. 18, p. 5103, Jan. 2020, doi: 10.3390/s20185103.

[57] N. Zhang, R. Bahsoon, N. Tziritas, and G. Theodoropoulos, "Knowledge Equivalence in Digital Twins of Intelligent Systems," *ACM Transactions on Modeling and Computer Simulation*, vol. 34, no. 1, pp. 1–37, Jan. 2024, doi: 10.1145/3635306.

[58] K. M. Alam and A. El Saddik, "C2PS: A digital twin architecture reference model for the cloud-based cyber-physical systems," *IEEE access*, vol. 5, pp. 2050–2062, 2017, Accessed: Jan. 08, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7829368/

[59] E. Ferko, A. Bucaioni, and M. Behnam, "Architecting Digital Twins," *IEEE Access*, vol. 10, pp. 50335–50350, 2022, doi: 10.1109/ACCESS.2022.3172964.

[60] J. Pfeiffer *et al.*, "Towards a Unifying Reference Model for Digital Twins of Cyber-Physical Systems." arXiv, Jul. 2025. doi: 10.48550/arXiv.2507.04871.

[61] J. Pfeiffer *et al.*, *Towards a Unifying Reference Model for Digital Twins of Cyber-Physical Systems*. 2025. doi: 10.48550/arXiv.2507.04871.

[62] N. Anwer, R. Stark, F. Tao, and J. Erkoyuncu, "Developing and leveraging digital twins in engineering design," *CIRP Annals*, vol. 2025, Jun. 2025, doi: 10.1016/j.cirp.2025.05.002.

[63] N. Anwer, R. Stark, F. Tao, and J. A. Erkoyuncu, "Developing and leveraging digital twins in engineering design," *CIRP Annals*, vol. 74, no. 2, pp. 843–868, Jan. 2025, doi: 10.1016/j.cirp.2025.05.002.

[64] M. Heithoff, N. Jansen, J. Michael, F. Rademacher, and B. Rumpe, "Model-Based Engineering of Multi-Purpose Digital Twins in Manufacturing," in *Digital Twin: Fundamentals and Applications*, S. Sabri, K. Alexandridis, and N. Lee, Eds., Cham: Springer Nature Switzerland, 2024, pp. 89–126. doi: 10.1007/978-3-031-67778-6_5.

[65] M. Heithoff, M. Trinh, J. Michael, B. Rumpe, and C. Brecher, "A Digital Shadow for Accurate Robot Motion Control: Integrating Data with Friction Models," Grand Rapids, USA, Jul. 2025. doi: 10.5283/epub.77667.

[66] M. G. Kapteyn, D. J. Knezevic, and K. Willcox, "Toward predictive digital twins via component-based reduced-order models and interpretable machine learning," in *AIAA Scitech 2020 Forum*, in AIAA SciTech Forum., American Institute of Aeronautics; Astronautics, 2020. doi: 10.2514/6.2020-0418.

[67] J. Liu, J. Vatn, and S. Yin, "A Case Study for Enhancing Maintenance of Cyber-Physical Systems With Digital Twin," *IEEE Transactions on Industrial Cyber-Physical Systems*, vol. 2, pp. 597–605, 2024, doi: 10.1109/TICPS.2024.3485038.

[68] E. Kamburjan, N. Bencomo, S. L. Tapia Tarifa, and E. B. Johnsen, "Declarative Lifecycle Management in Digital Twins," in *Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems*, Linz Austria: ACM, Sep. 2024, pp. 353–363. doi: 10.1145/3652620.3688248.

[69] I. Aldalur, A. Arrieta, A. Agirre, G. Sagardui, and M. Arratibel, "A microservice-based framework for multi-level testing of cyber-physical systems," *Software Quality Journal*, vol. 32, no. 1, pp. 193–223, Mar. 2024, doi: 10.1007/s11219-023-09639-z.

[70] G. Lyan, J.-M. Jézéquel, D. Gross-Amblard, and B. Combemale, "DataTime: A Framework to smoothly Integrate Past, Present and Future into Models," in *2021 ACM/IEEE 24th International Conference on Model Driven Engineering Languages and Systems (MODELS)*, Oct. 2021, pp. 134–144. doi: 10.1109/MODELS50736.2021.00022.

[71] C. Ellwein *et al.*, "Rethinking Asset Administration Shell Communication Types: A Systematic Mapping Study and Portfolio-Based Classification," *Production Engineering*, vol. 20, Dec. 2025, doi: 10.1007/s11740-025-01378-3.

[72] A. J. G. de Azambuja, T. Giese, K. Schützer, R. Anderl, B. Schleich, and V. R. Almeida, "Digital Twins in Industry 4.0 – Opportunities and challenges related to Cyber Security," *Procedia CIRP*, vol. 121, pp. 25–30, Jan. 2024, doi: 10.1016/j.procir.2023.09.225.

[73] E. Katsoulakis *et al.*, "Digital twins for health: A scoping review," *npj Digital Medicine*, vol. 7, no. 1, p. 77, Mar. 2024, doi: 10.1038/s41746-024-01073-0.

[74] A. Osipov *et al.*, "The Molecular Twin artificial-intelligence platform integrates multi-omic data to predict outcomes for pancreatic adenocarcinoma patients," *Nature Cancer*, vol. 5, no. 2, pp. 299–314, Feb. 2024, doi: 10.1038/s43018-023-00697-7.

[75] S. Sabri, K. Alexandridis, and N. Lee, Eds., *Digital Twin: Fundamentals and Applications*. Cham: Springer Nature Switzerland, 2024. doi: 10.1007/978-3-031-67778-6.

[76] Q. Xu, T. Yue, S. Ali, and M. Arratibel, "Pretrain, Prompt, and Transfer: Evolving Digital Twins for Time-to-Event Analysis in Cyber-physical Systems." arXiv, Apr. 2024. doi: 10.48550/arXiv.2310.00032.