The Cost of Functional Safety

Why the industry’s largest failure mode is also its cheapest to prevent — and how to find out what it is costing you

Richard Kelly CFSE, MSc CEng (IET)
ROAK Engineering Ltd
September 2026

Every organisation delivering safety instrumented systems spends money on functional safety. That money goes to one of two places. It either prevents an error, or it corrects one.

Almost no organisation knows the split.

Summary

Every organisation delivering safety instrumented systems spends money on functional safety. That money goes to one of two places. It either prevents an error, or it corrects one.

Almost no organisation knows the split, because correction cost is never coded to functional safety. It is absorbed into project hours and reported as overrun, where it looks like the job was difficult.

This paper argues three things.

First, that functional safety errors are structurally more expensive than ordinary engineering errors, because the safety lifecycle is a chain of dependent phases and an error propagates into everything built on top of it.

Second, that the industry's dominant failure mode is created at the point where correction is cheapest. HSE's incident analysis in *Out of Control* found 44% of control system failures had their primary cause in an inadequate specification. Reading the underlying table more closely produces a sharper finding: only 12% concerned functional requirements. **Thirty-two per cent concerned safety integrity requirements** — the SIL targets, failure measures, proof test assumptions and architectural constraints that a functional safety management system (FSMS) exists to govern.

Third, that the industry's response to this is systematically the wrong one. Assessment, audit, verification and independent review are all detection activities. They occur after the decision that caused the problem. An assessment does not make an organisation safe. It reports whether it was.

The paper closes with a practical instrument for measuring correction cost in a delivery business, using records the business almost certainly already keeps.

---

1. A cost nobody codes

Ask an engineering business what functional safety costs them and you will get one of two answers. Either a number that covers procedures, training and assessment, or a shrug.

Neither answer includes the recalculated SIL study. Or the safety requirements specification reissued at Revision C because the requirements were never formally agreed. Or the hazard study reopened halfway through detailed design. Or the site acceptance test re-run because the test was written against the installation rather than against the requirement. Or the fortnight two engineers spent assembling three years of evidence for a client audit at a week's notice.

None of that is coded to functional safety. It is booked as engineering hours against the project. When the project overruns, the overrun is attributed to scope, or to the client, or to the job simply being a difficult one.

If you are the technical director who signs that number off, or defends it when a client asks, this is the part that should concern you. The shrug is not because your engineers are hiding the cost from you. It is because nothing in the business is set up to show it to anyone, including them.

This is not an accounting quirk. It is the central problem, and it has a direct consequence:

> ***An organisation cannot manage a cost it cannot see.***

The cost of preventing functional safety errors is visible, budgeted and easy to cut. The cost of correcting them is invisible, unbudgeted and therefore never cut. Any organisation optimising on what it can see will systematically underfund prevention and overspend on correction, and will experience the result as bad luck.

---

2. Why functional safety errors behave differently

The argument that prevention costs less than correction is not new. It was made for manufacturing by Philip Crosby in *Quality is Free* in 1979, and versions of it appear across quality and systems engineering literature.

What is specific to functional safety is *why* the difference is so large. It follows from the structure of the lifecycle rather than from any general principle.

The safety lifecycle is a chain of dependent phases. Each phase consumes the output of the previous phase as its input. The hazard and risk assessment feeds SIL determination. SIL determination feeds the safety requirements specification. The specification feeds design. Design feeds the application program. All of it feeds validation and test.

That dependency is what makes a functional safety error behave unlike an ordinary engineering error. It does not stay where it was made. Everything built afterwards inherits it.

Consider a single missed scenario:

> *The hazard study does not identify it. SIL determination never sees it, so no safety instrumented function is allocated. The safety requirements specification does not*

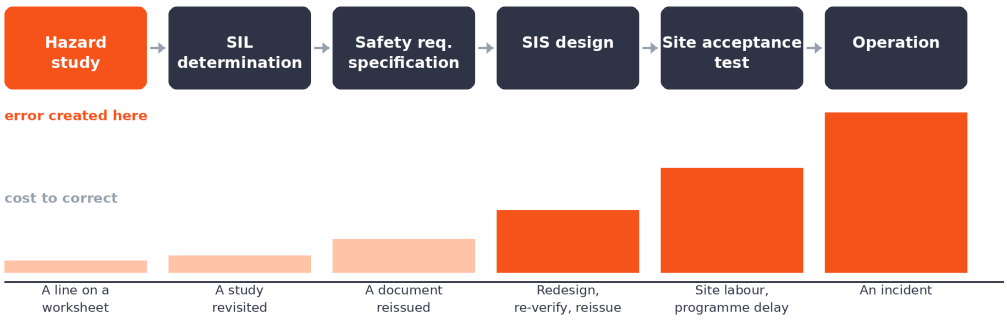
> *specify it. The design does not implement it. Site acceptance testing cannot test for something the specification never required. It surfaces in operation, as a demand the system does not answer.*

Now price the correction at each point in that chain:

Discovered at	Correction required
Hazard study	A line added to a worksheet
Safety requirements specification	A document reissued
Design review	Redesign, re-verification, reissue
Site acceptance test	Site labour, programme delay, possibly hardware
Operation	An incident

Figure 1 — How one missed scenario becomes expensive

An error in a dependent chain is inherited by everything built after it



The escalation is not a rule of thumb. Correcting the error means rebuilding everything built on top of it.

Figure 1 — How one missed scenario becomes expensive

The escalation is not a slogan or a rule of thumb. It is arithmetic. Correcting an error in a dependent chain means rebuilding everything that was built on it, and the further down the chain the discovery, the more there is to rebuild.

It is worth being precise about a figure often quoted alongside this argument. The familiar 1:10:100 escalation ratio — one unit of cost at requirements, ten at design, a hundred after release — originates in Barry Boehm's cost-to-fix data assembled at TRW with corroborating data from IBM, GTE and Bell Labs, first presented in 1976 and published in *Software Engineering Economics* in 1981. It is a software engineering finding. What's well supported is the shape of that curve — cost rises sharply the later an error is caught. The specific 10x and 100x multipliers are Boehm's software data, not measurements from a process plant, which is exactly why the next section uses different evidence — HSE's own incident analysis of control systems.

3. The evidence

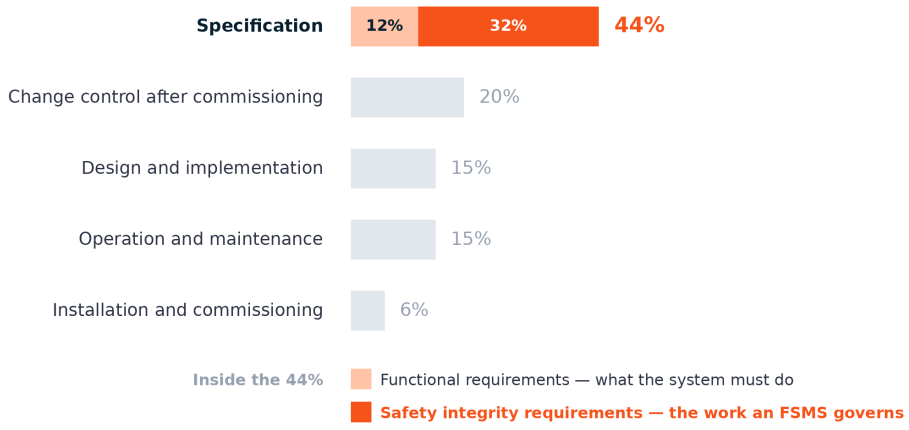
In 1995, and again in a second edition in 2003, the UK Health and Safety Executive published *Out of Control: Why control systems go wrong and how to prevent failure* (HSG238). It analysed 34 incidents involving control systems, identifying 56 causes, and classified the primary cause of each against the lifecycle phase in which it originated.

The published table:

Primary cause by phase	Frequency	%
Inadequate functional requirements specification	4	12
Inadequate safety integrity requirements specification	11	32
Total inadequate specification	**15**	**44**
Total inadequate design and implementation	5	15
Total inadequate installation and commissioning	2	6
Total inadequate operation and maintenance	5	15
Total inadequate change control after commissioning	7	20

Figure 2 — Primary cause of control system failure, by lifecycle phase

HSE, Out of Control (HSG238) — 34 incidents, 56 causes identified



The 44% is widely quoted. The split inside it is not: only 12% concerned what the system was required to do.

Nearly three quarters of specification failures were failures of the integrity requirement.

HSE note the sample is small and of low statistical significance (HSG238, para 74).

Figure 2 — Primary cause of control system failure, by lifecycle phase

The headline is widely cited. **Forty-four per cent of control system failures originate in the specification** — the first phase of the lifecycle, the cheapest phase to change, and the one furthest from the point at which failure becomes visible.

3.1 The finding inside the finding

The headline figure is quoted far more often than the two rows that produce it, and those rows say something more specific.

Only **12%** of primary causes were inadequate *functional* requirements specification — the organisation not correctly establishing what the system was required to do.

Thirty-two per cent were inadequate *safety integrity* requirements specification.

That is a different problem with a different cause. It is not that engineers fail to understand the process or misunderstand the required action. It is that the integrity requirement is wrong: the SIL target, the required failure measure, the architectural constraints, the proof test interval and the assumptions underpinning it.

Every one of those is a governed output of a functional safety management system. Setting a SIL target requires a documented determination method and a justification for choosing it. A target failure measure requires traceable failure data. Architectural constraints require a stated and justified hardware fault tolerance route. A proof test interval requires the assumption made in verification to be carried through to the maintenance schedule.

Where there is no management system, none of those become wrong through incompetence. They become wrong through informality — decided reasonably, by capable people, without a defined method, and without a record of the reasoning that would let anyone check.

The largest identified cause of control system failure is precisely the work that a functional safety management system exists to govern, and precisely the work that is done on judgement alone when there is no system.

3.2 The limitation, stated plainly

HSE are explicit about the strength of this data. Paragraph 74 of the report:

> *"It is acknowledged that because of the small sample size the results of the analysis have low statistical significance, and therefore care needs to be taken in using these results to generalise for all control system failures."*

Thirty-four incidents is a small sample and should be described as one. The direction of the finding, however, is not seriously disputed, and HSE note at paragraph 76 that studies in software development independently show that specification errors account for most faults and failures.

The claim this paper makes is therefore a modest one. Not that 44% is a precise industry constant, but that specification is where most failures begin, and that this is the phase in which correction is close to free.

---

4. Why the industry's response makes it worse

Consider what functional safety spends its professional energy on. Functional safety assessment. Audit. Verification. Independent review. Site acceptance testing. Validation.

All of these are necessary. Several are required by IEC 61511. None of them prevent anything.

Every one is a detection activity, and every one occurs after the decision that created the problem. They differ only in how long afterwards. Verification examines a phase once its outputs exist. Assessment forms a judgement on work already done. Testing exercises a system already built.

> ***An assessment does not make an organisation safe. It reports whether it was.***

This is not an argument against assessment. Functional safety assessment is required, it is the only mechanism that brings genuinely independent judgement to bear, and IEC 61511-1 Clause 5.2.6.1.2 is right to require a senior competent person from outside the design team.

It is an argument about what detection can and cannot do. If the specification was wrong, assessment finds it late — at the point in the dependent chain where correction is most expensive and where the programme has least room to absorb it. The finding is correct and the cost of acting on it is maximal.

The management system is what makes the work right in the first instance. Assessment confirms it. An industry that invests heavily in assessment and lightly in the management system should expect to keep paying at the wrong end.

---

5. The Cost of Functional Safety model

The argument so far is diagnostic. What follows is the model that makes it usable: two ledgers that describe where the money goes, two levers that move it, and one measure that tells an organisation whether the levers are working.

5.1 Two ledgers

It helps to name the two halves in language that survives a project meeting.

The cost of getting it right. Planning. Requirements defined and agreed before work begins. Competence established before assignment. Verification at each phase boundary. This cost is visible, predictable and can be budgeted. It falls as an organisation matures, because the work stops being reinvented on every project.

The cost of putting it right. Recalculating a SIL study because the failure data could not be traced to a source. Rewriting a specification nobody agreed. Reopening a hazard study during detailed design. Re-running a site acceptance test. Assembling evidence at short notice for a client audit. And, less visibly, the tenders that were not won because there was nothing to show for them.

This second cost is invisible by default. It is not that organisations conceal it — there is simply no account for it to be posted to.

Figure 3 — The same money, spent at two different ends

Only one of the two appears in a budget

Every organisation delivering SIS already pays one of these. The only decision is which.

Figure 3 — The same money, spent at two different ends

Crosby's original terms for these were the price of conformance and the price of nonconformance. The plainer names travel further in an engineering business, and travelling is the point: a cost that cannot be named in a project meeting will not be managed in one.

5.2 Two levers

Naming the ledgers does not move anything between them. What moves the money is the observation in section 2, stated as a rule:

> ***Correction cost is a function of how far an error travels before anyone catches it.***

That follows directly from the dependent chain. The severity of an error is not primarily a property of the error. A missed scenario, a wrong SIL target and an untraceable data source all cost roughly nothing to fix in the phase that produced them, and all cost a great deal by the time the system is on site. What changes between those two outcomes is distance.

If that is true, then there are only two ways to reduce correction cost, and every activity in a functional safety management system is one or the other.

Lever one — fewer errors created. Requirements agreed and baselined before work starts. Competence established before assignment rather than inferred from experience. A defined method with traceable data behind every integrity requirement. This lever attacks the 32%.

Lever two — shorter distance travelled. Verification at each phase boundary, so that an error surfaces in the phase that created it rather than three phases downstream. This lever does not reduce the number of errors at all. It reduces what each one costs.

The second lever is the one organisations neglect, because it produces no visible output. A phase gate that finds nothing looks like time wasted. It is the mechanism preventing the expensive half of the problem.

Figure 4 — The Cost of Functional Safety model

Correction cost is a function of how far an error travels before anyone catches it

IMPACT AT A GLANCE			
Average distance travelled	2.5 → 0.5 phases	80%	shorter
Errors created	4 → 2	50%	fewer

THE MEASURE

Average phases travelled before an error is found.

It falls before total hours do, which makes it the earliest evidence that a management system is working.

Fewer errors created. Shorter distance travelled. Lower cost. Safer systems.

5.3 One measure

A model that cannot be measured is an opinion held confidently. This one has a single natural metric:

> ***The average number of lifecycle phases an error travels before it is found.***

Zero means errors are being caught in the phase that made them, which is the target state. It requires no rates, no cost allocation and no agreement about what an hour is worth. It is calculated from two fields that any delivery organisation can record: the phase in which the error was created, and the phase in which it was found.

It also has a useful property. It falls before total correction hours do — because verification starts catching errors earlier some time before the organisation stops making them. That makes it the earliest available evidence that a management system is having an effect, at the point in an implementation where evidence is most needed and least available.

Alongside it sits a second figure worth tracking: the proportion of correction hours originating in the specification phases, which can be compared directly against the HSE distribution in section 3.

5.4 Where an organisation sits

The two levers describe direction. Maturity describes position, and organisations move through recognisable stages:

Stage	What is true
Uncertainty	Nothing is set up to prevent error. Every project rediscovers the same requirements. Correction cost is total and invisible.
Awakening	Prevention is understood to be cheaper. Nothing is funded. Procedures appear when a client asks and go quiet when they stop.
Enlightenment	Some phases are covered and cheaper to run. The rest are still corrected after the fact. The split becomes visible.
Wisdom	Prevention does most of the work. Errors surface in the phase that created them. What remains is usually handover documentation.
Certainty	Work goes out right first time because requirements are clear before anyone starts. Independent assessment confirms rather than discovers.

The stage names are Crosby's, from the maturity grid in *Quality is Free*. They are used here because they describe the progression accurately and because inventing new labels for a well-described sequence adds nothing.

Position on that scale is not a judgement about engineering ability. It is a statement about which end of the lifecycle an organisation currently pays at.

---

6. What a management system actually does

The description of a functional safety management system as “documentation, in case someone asks” is common, and it is the reason such systems are so often bought late, resented, and left unused.

A management system is a prevention mechanism. Four things it does, each attacking a specific and identifiable error class — and each operating through one of the two levers.

The requirement exists before the work starts. This attacks the 44%, and specifically the 32%. If the safety requirements specification is agreed and baselined before design begins, and if the integrity requirements within it are produced by a defined method with traceable data, the single largest identified source of failure is closed off at the point where closing it costs almost nothing.

Verification sits at the phase boundary. IEC 61511-1 Clause 6.3.3 requires that each phase for which safety planning has been carried out is verified in accordance with Clause 7. The commercial reading of that requirement is the important one: an error caught in the phase that created it costs hours, and the same error caught two phases later costs weeks, because everything built on it must come apart. Phase-gate verification is not administrative overhead. It is the mechanism that stops propagation.

Competence is established before assignment, not inferred from experience. A capable general engineer producing plausible-looking incorrect work is the most expensive failure mode available, because it survives casual review. Deciding competence deliberately against the specific activity, and recording the reasoning, is what prevents it. IEC 61511-1 Clause 5.2.2 requires this for all persons involved in lifecycle activities, and requires the competence to be reassessed periodically and on change of role.

Evidence accumulates as a by-product. Where work is performed in the defined order and recorded as it goes, evidence exists already when it is needed. The assessment stops being an event that must be prepared for, and nobody is removed from billable work to reconstruct a history.

Only the fourth of these has any direct connection to being assessed — and even that one is better understood as recovered capacity than as compliance.

---

7. What it is worth

Fix this, and three outcomes follow for your business, in descending order of size and ascending order of visibility.

Correction cost stops leaving the business. This is the largest of the three and the least visible. It is money already being spent, on work already being done twice, currently reported as something else.

Larger work becomes winnable. Being able to evidence functional safety arrangements before being asked clears prequalification and tender gates that price alone will not open. Organisations without a system are quietly excluded from work they are technically capable of delivering, and are rarely told that this was the reason.

Risk stops being carried unnecessarily. Not regulatory risk in the abstract, but the specific and personal exposure of having approved a SIL claim whose basis cannot now be reconstructed, on a plant still in operation, some years after the engineer who produced it moved on.

---

8. The limits of the argument

Three qualifications, offered because an argument is more useful when its boundaries are known.

Conformance assumes the requirement is correct. The quality-world formulation is conformance to requirements. In functional safety the requirement itself can be wrong — a missed hazard is exactly that, and it accounts for part of the 44%. Prevention here must therefore include challenging the requirement, not merely conforming to it, which is why hazard study and specification outputs require independent review. In this respect functional safety is more demanding than manufacturing quality, not less.

Zero defects fails as an exhortation. W. Edwards Deming attacked the idea directly; his tenth point calls for the elimination of slogans and exhortations asking the workforce for zero defects, on the grounds that most causes of poor quality are properties of the system and lie beyond the workforce's power to correct. He was right. Applied to people, zero defects is a slogan and it fails. Applied to the system, it is a design standard and it works. The distinction is not cosmetic.

The measurement problem is real and unsolved in most organisations. Everything above is an argument that correction cost is large. Very few organisations can produce a number, and until they can, the argument remains a conviction rather than a finding. That is the subject of the final section.

---

9. Measuring it

The instrument required is modest. It is a register that records, for each instance of functional safety work that had to be done again, six things:

- what was redone
- which record it came from
- **the phase in which the error was created**
- **the phase in which it was found**
- the engineering hours spent
- any direct cost beyond those hours

The two emphasised fields carry most of the value. The distance between them is how far the error travelled before anyone caught it, and that distance is the direct measure of whether phase verification is working. It generally improves before total hours do, which makes it the earliest available evidence that a management system is having an effect.

Three practical points determine whether such a register survives contact with a real project.

It must not create new data collection. Almost every entry already exists somewhere — as a failed check on a verification checksheet, a finding in the action item log, a management of change record, a punch item from site acceptance testing, or a client review comment. The register is a roll-up of records already kept. Presented as a new obligation it will be abandoned within a quarter.

It should record hours, not money. Rates are contentious, vary by grade and change annually. Hours do not. Conversion to money can happen at the point of reporting, by whoever is entitled to apply a rate.

It must include the small entries. Work redone in an afternoon, which nobody would have called rework, is what makes the total honest. Organisations that record only the disasters conclude that they have very little correction cost, which is precisely the conclusion the exercise exists to test.

Running this against a couple of your own recent projects, rather than starting from a blank page, is exactly what the diagnostic call does — we do it together, live on the call.

Once a period of data exists, the organisation can compare its own distribution of failure origins against the HSE distribution — and can, for the first time, state what functional safety is actually costing it, in both halves.

---

10. Conclusion

Functional safety is not a cost that an organisation can decide whether to incur. It is a cost every organisation delivering safety instrumented systems already carries. The only decision available is which end of the lifecycle to carry it at.

Carried at the front, it appears as planning, defined requirements, established competence and verification at the phase gates. It is visible, it is budgeted, and it reduces over time.

Carried at the back, it appears as recalculation, reissue, re-test, site revisits, evidence assembled under pressure and work lost to competitors who could demonstrate what they do. It is invisible, it is unbudgeted, and it does not reduce, because nothing in the organisation is aimed at it.

The available evidence indicates that most of the failures worth preventing are created at the very start of the lifecycle, in the phase where prevention is nearly free — and that within that phase, the integrity requirements rather than the functional requirements account for most of the problem.

That is a solvable problem. It is solved by a management system, and it is solved at the front.

Next step

The argument in this paper is diagnostic. Acting on it starts with one number: where your organisation currently sits between the two ledgers in section 5, and how far your own errors are travelling before anyone catches them.

That number is what the Functional Safety Diagnostic Call establishes. It is a working session, not a sales call. We go through what you already have in place, find the gaps before they cost you a tender or a rebuild, and give you a clear picture of which end of the lifecycle you are currently paying at.

Book Your Functional Safety Diagnostic Call

Book a call →

I run these calls personally — two a week, so each one gets proper attention.

References

Health and Safety Executive. *Out of Control: Why control systems go wrong and how to prevent failure.* HSG238, 2nd edition, 2003. Table 2 and paragraphs 72–76.

Crosby, P. B. *Quality is Free: The Art of Making Quality Certain.* McGraw-Hill, 1979.

Boehm, B. W. *Software Engineering Economics.* Prentice Hall, 1981.

Deming, W. E. *Out of the Crisis.* MIT Press, 1986. Point 10.

BS EN 61508:2010, Parts 1–7. *Functional safety of electrical/electronic/programmable electronic safety-related systems.*

BS EN 61511-1:2017+A1:2017. *Functional safety — Safety instrumented systems for the process industry sector.* Clause 5.2.2, Clause 5.2.6, Clause 6.3.3, Clause 7, Clause 10.

ROAK Engineering Ltd provides functional safety management systems and independent assessment support to engineering delivery organisations. This paper may be quoted with attribution.