SHAPE, SELECT, OPTIMISE

WHITE PAPER

pFAST

How likely is a process with embedded AI to function as intended? A risk based framework for the design of processes

probabilistic Function Analysis System Technique

Stephen Cresswell

SHAPE SELECT OPTIMISE

Independent value, risk and innovation analysis

June 2026

Abstract

There is widespread recognition of the opportunity that AI offers to many organisations, and the inevitability that adaptation must take place. However, capturing this opportunity presents a step into the unknown. Principled methodologies are needed to decide what should remain as human discernment, what to automate, and how to manage the uncertainty of a rapidly changing technology that is uncertain by nature.

Function Analysis is a long established methodology that considers what products and process 'do' not what they 'are'. The obvious connection to AI is that agents 'do' things; performing functions on behalf of the designer. With a better understanding of what we want agents and machines to do, we can express our instructions in terms of outcomes that are valuable to us. There is also a long history of frameworks and methodologies for managing risk and uncertainty in complex safety critical systems. The success of these frameworks is evidenced by the increase in safety over decades.

This paper introduces **pFAST** (probabilistic Function Analysis System Technique), an extension of established function analysis, adapted for the age of AI. pFAST provides a structured method for decomposing processes into their constituent functions, assessing the risk profile of each function using the **CARBS** framework (Criticality, Assessability, Rectification, Base rate, Stability), and making principled decisions about human versus machine execution. The approach is intended to offer organisations a solution to design AI-augmented processes that are assured, auditable, future proof and cost-effective.

Base rate — the prior probability that a given function produces a correct output — is a quantitative dimension not found as an explicit input in the published AI risk frameworks surveyed for this paper. It assesses the anticipated reliability or success rate of the function at the time of design. Its inclusion directly addresses the well-documented cognitive trap of base rate neglect, identified in 'Thinking, Fast and Slow' as a primary driver of overconfident assessments.

© 2026 Shape Select Optimise — Stephen Cresswell

What is Function Analysis?

Function analysis is the discipline of identifying and describing what an object or process does functionally. Its purpose is to strip away assumptions about what something *is* and consider what it *does* so that alternative, potentially better and more efficient solutions can be identified.

Function Analysis has its origins in General Electric's procurement function during the Second World War. Shortages led to suppliers being asked to offer alternatives that could *function* in the same way as rubber, steel and other materials that were unavailable. Function Analysis was codified in the 1960's by Charles Bytheway[1]. Since then function analysis has been an underlying discipline for value engineering and value management. Note that the term "value engineering" has acquired a separate meaning in the AI field, referring to the alignment of AI systems with human values and ethics. Function analysis is also an underlying discipline of TRIZ[2], a method for inventive problem solving that draws on established patterns.

Central concepts in function analysis are function definition and How/Why logic. Individual Functions are codified as verb-noun pairs. For instance, the basic function of a road bridge is 'Span Gap'. It has other functions like 'Look Good' (an aesthetic function) and 'Separate Traffic' (safety function). The functionality of a process, product or object can be expressed by listing its functions and sub functions using verb-noun pairs.

The way in which functions work together as a system can be shown graphically in a FAST (Function Analysis System Technique) diagram. This displays the hierarchy of functions in a tree structure. The hierarchy is derived from 'How/Why' Logic. Working from left to right the logic is 'how' is this function performed? Then from right to left 'Why' is this function performed.

[1] FAST Creativity & Innovation - Google Books

[2] TRIZ (Teoriya Resheniya Izobretatelskikh Zadach, the Theory of Inventive Problem Solving) is a systematic innovation methodology, developed by Soviet engineer Genrich Altshuller from his analysis of hundreds of thousands of patents, which holds that inventive problems and their solutions recur across industries, so they can be solved deliberately by resolving the underlying technical contradictions using a known set of inventive principles rather than relying on chance or trial and error.

© 2026 Shape Select Optimise — Stephen Cresswell

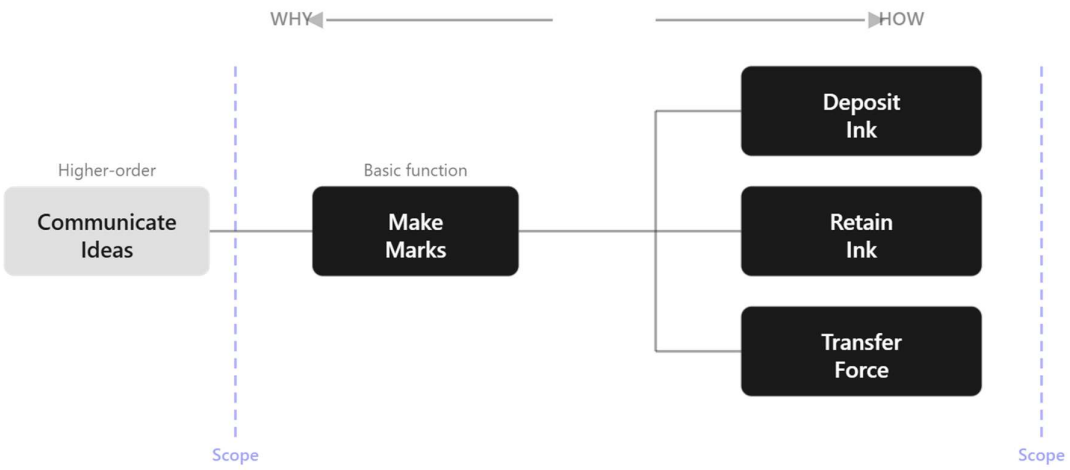


Diagram 1 - Function Analysis System Tree for a Pen

Although it can be explained simply, function analysis can be a challenging and intellectually intensive exercise. This is because it creates deep reflection on the product or process being analysed. Many processes and products incorporate deeply embedded assumptions and perspectives that are no longer relevant.

Probability and Uncertainty

Historically, the subjects of function analysis were, or would often be treated as, deterministic and objective systems, whether physical devices or software code. In practice, there were often process functions performed by people that were both subjective and probabilistic — for instance, “Summarise Change-Request” might produce different outputs from different people given the same input. However, this uncertainty was often accepted rather than explicitly managed.

Functions fulfilled by AI are probabilistic in nature: the system may not perform with 100% accuracy, and it may produce different responses to the same input on different occasions. They are not only probabilistic in the moment, but they are also subject to change over time. Models evolve and adapt, and human behaviour is changing in response to the introduction of new technology.

The introduction of AI has brought probabilistic characteristics into sharp focus as a design consideration. When a function is performed by a large language model, the designer must account for variability, hallucination, context-sensitivity and the absence of guaranteed outcomes. This is, in essence, a risk and uncertainty management challenge.

© 2026 Shape Select Optimise — Stephen Cresswell

Lusser's Law: "Weaker than the weakest link"

In order to achieve overall system reliability, reliability of individual functions needs to be higher than intuitively might be expected. This principle has been understood since the 1950s, when German engineer Robert Lusser formalised it as Lusser's Law during rocket component reliability analysis. Recent academic work[3] has confirmed its applicability to AI systems; error compounding across multi-step and multi-agent workflows may be one of the central reliability challenges of the AI era.

The table below shows the effect of this reliability/error compounding.

	Number of Function Nodes			
Individual Function Reliability	2	5	10	20
99.90%	99.8%	99.5%	99.0%	98.0%
99%	98.0%	95.1%	90.4%	81.8%
95%	90.3%	77.4%	59.9%	35.8%
90%	81.0%	59.0%	34.9%	12.2%
80%	64.0%	32.8%	10.7%	1.2%
70%	49.0%	16.8%	2.8%	0.1%

Table 1 – Overall system reliability

For instance, a process with five individual functions, each of which is 95% reliable, results in only a 77% reliability of the process overall. This assumes that the individual functions are independent, it may also be the case that errors compound in a non-linear manner making matters worse. This is straightforward compound probability: $0.95^5 = 0.774$.

Probabilistic Function Analysis System Technique

pFAST (probabilistic Function Analysis System Technique) extends the classic FAST by explicitly incorporating uncertainty assessment into the functional decomposition. Each function in the tree is not only defined and positioned in the How/Why hierarchy, but also assessed for its risk profile if it is executed by a probabilistic agent.

Overall system reliability can be predicted by multiplying the reliability of each individual function node. This prediction can then be an integral part of the design development, helping to ensure that the design fulfils the reliability performance required. Once deployed, the performance of the system can be monitored to verify whether the predicted reliability is realised. In turn, this can be used as quantitative diagnostics to fix and resolve system issues.

The risk framework for assessing the nodes is called CARBS.

[3] E.g. Rabanser et al. ("Towards a Science of AI Agent Reliability," arXiv:2602.16666, 2026)

© 2026 Shape Select Optimise — Stephen Cresswell

The CARBS Risk Assessment Framework

Once functionality is understood, each automated function in the pFAST tree is assessed against five dimensions, grouped as CARBS: Criticality, Assessability, Rectification, Base Rate[4], and Stability. The first four (CARB) assess the function at a point in time; the fifth (S) is reflective and asks how long that assessment will remain valid. Together, these determine whether an individual function should be assigned to human discernment, an AI agent, a deterministic script, or some other solution.

CARBS does not claim radical conceptual novelty — it is a selection and reframing of established reliability & risk frameworks. Mostly, these frameworks have been used in applications where very high reliability is required, for instance, avionics and railway signaling. The general business and administration context is different; a lower level of reliability is acceptable. If a totally manual process was replaced with an automated process that was only 50% reliable but always 'failed safe' and reverted to a manual process, this could still be valuable from a productivity perspective. CARBS is calibrated for this reality: it asks not "can this be made perfect?" but "what level of reliability is needed here, and how will we know when we fall short?"

AI presents unique challenges compared to the deterministic systems for which classical reliability engineering was developed: adaptability, inscrutability, and emergent behaviours mean that AI cannot be treated as a complex but predictable component. The closest structural analogue is FMEA (Failure Mode and Effects Analysis), which similarly uses Severity, Detectability, and Occurrence dimensions. However, CARBS differs in three important ways: it operates at the AI function level rather than the mechanical failure mode level; it introduces Base Rate as a discrete prior-probability dimension (reframed from FMEA's failure-focused Occurrence into a positive, function-level prior); and it adds Stability as a time dependent meta-dimension.

[4] Base rate fallacy - Wikipedia

© 2026 Shape Select Optimise — Stephen Cresswell

Dimension	Description	Notes
Criticality	How important is it that this function is performed correctly? Does the function 'fail safe', 'fail dangerous' or is it 'intrinsically safe'.	Appears in Failure Mode and Effects Analysis (FMEA) & Failure Mode Effects and Criticality Analysis (FMECA). HAZOPs.
Assessability	Can it be determined, quickly (before follow on action) and cheaply, whether the function was performed correctly?	Analogous to audit and control effectiveness assessments (e.g. ISO 9001).
Rectification	How easily and at what cost can a mistake be corrected? or a situation recovered?	Maps closely to FMEA's "Detectability" and "Severity" dimensions.
Base Rate	With what frequency can the function be expected to perform correctly? Which solution offers the highest reliability?	Directly borrowed from statistics and Bayesian reasoning. Base rate neglect — the tendency to ignore prior probabilities when making judgements — is cited in Kahneman's Thinking, Fast and Slow as a primary driver of overconfident assessments.
Stability	How long will the CARB assessment of this function remain valid? A meta-dimension that sits above the other four, assessed at both function level (is this function's risk profile liable to change?) and system level (does the process remain stable as AI and human behaviour co-evolve?). The practical output is a reassessment trigger: a recommended frequency or event at which the CARB analysis should be re-run.	No direct equivalent. Unlike CARB dimensions (which assess static properties), Stability addresses temporal validity — the recognition that AI systems drift, update, and alter the human behaviour around them. High Stability = long reassessment interval. Low Stability = frequent re-evaluation required. Instrumentation (token cost, agreement rate, override frequency) provides empirical early warning of degrading Stability.

© 2026 Shape Select Optimise — Stephen Cresswell

Intended Benefits of pFAST for Technology Transformation

The central challenge with the implementation of AI is how to capture the opportunity it presents whilst also managing the risks. These risks take a number of forms, failed implementation risk, cost risk - either in development or during its operation. The anticipated benefits of using Function Analysis and quantitative risk assessment in the AI technology transformation:

Clarity of Purpose - The higher-order function crystalised during the FAST analysis is what the user or organisation actually wants the process or product to do. The discipline of Function Analysis pushes the designer to decompose functions thoughtfully in pursuit of the higher order function. This means better expression of the designer's intent and fewer gaps for the AI to fill in, which the AI will often do through generic patterns recognised in the training data.

Appropriate Division of Labour and Solutions - Each function node on the FAST tree can be considered for: human discernment, expert judgement, agent-with-tool, sub-agent or deterministic script. This provides a principled alternative to the current ad-hoc "string some prompts together" approach. Potentially, the functional blueprint provides evidence and traceability that the process was 'designed to function right' in regulatory contexts or where design decisions need to be reviewed by others.

Increased likelihood of success - explicit and quantitative management of the risk and uncertainty involved from a functional perspective increases the likelihood solutions wil do what is intended. Fundamentally, this is a risk based approach – assigning executors based on appreciation of reliability and uncertainty.

Future Proofing - If the specification is functional ("validate invoice", "schedule site-visit", "summarise risk-register"), the specification outlives the implementation. This seems to be one of the biggest practical problems in AI adoption— organisations write agent requirements in terms of today's tools, and the specifications are outdated within weeks or months. Over time the functions can be reviewed and then updated as required.

Value engineering and LEAN Operations for AI deployment. Classic value engineering sought the cost-to-perform each basic function compared to the benefit it brings. In the modern agent context, we now have hard numbers — tokens, API calls, latency, error rate per function.

© 2026 Shape Select Optimise — Stephen Cresswell

FAST Brainstorming Application - A Real Implementation

To demonstrate pFAST in practice, consider the creation of a Function Analysis Brainstorm App — a tool that coaches users through the first stage of creating a FAST diagram: brainstorming and validating functions. The application is live at functionanalysis.app. The app reviews brainstormed items, filters out items that are not functions, and suggests modifications to achieve the verb-noun pair format. The process can be summarised as follows:

The basic function of the brainstorm stage is ‘create function-list’

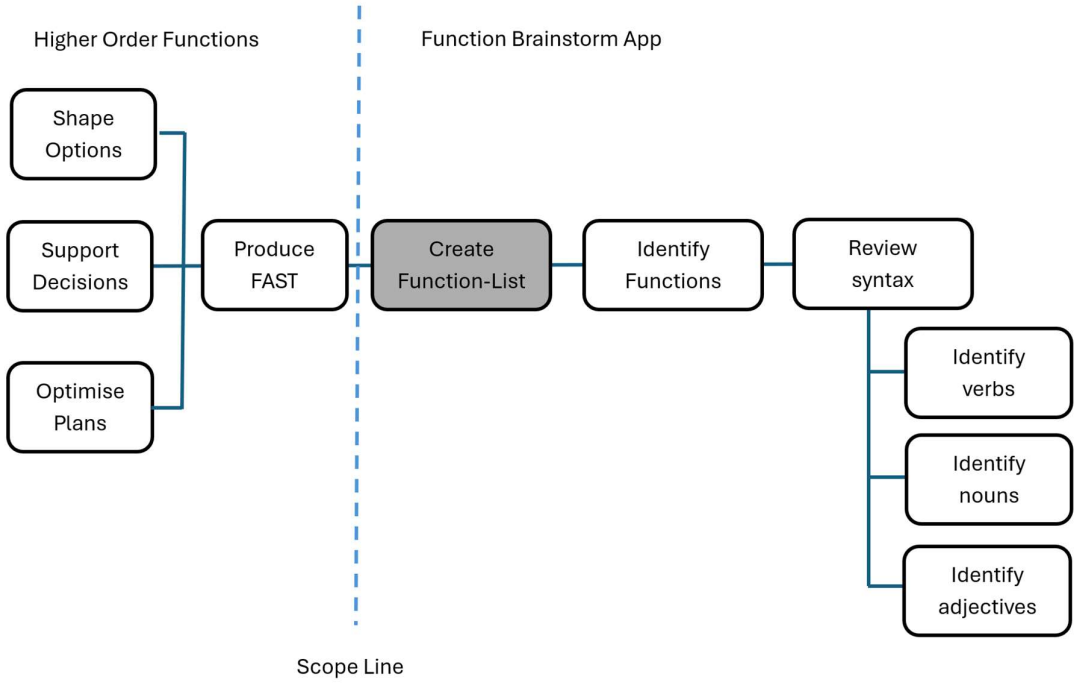


Diagram 2 - Function Analysis System Tree for function brainstorm app

Applying the CARB framework to the core function “Review Function-Syntax”:

Dimension	Assessment
Criticality	Medium. Incorrect syntax review leads to poorly formed functions in the FAST diagram, but the consequences are confined to the quality of the brainstorm — no safety, financial or legal impact. As a coaching app, all syntax is reviewed by humans anyway.
Assessability	Mixed. Verb form and hyphenation can be checked programmatically. However, determining whether a noun is “too vague” or whether an item is a function versus a design objective requires human judgement. The AI can flag potential issues but cannot definitively resolve ambiguous cases.
Rectification	Low cost. The user can override any classification, edit any function text, and move items between categories at any point. The coaching model means no AI decision is final — the human always has the last word.
Base Rate	Estimated 65–75% unassisted accuracy. Clean verb-noun pairs are classified correctly ~95% of the time, but verb-noun homographs (words like “fire”, “support”, “process” that can be either verb or noun) drop accuracy to 57–75%.
Stability	High stability. The chosen models are fixed in the workflow. The application use case is also stable.

Claude was prompted to estimate the reliability of the process as initially designed. This revealed a relatively low base rate of success: 65 – 75%. Since the process was designed as a coaching app there would always be a human in the loop, and for this reason it was relatively low risk. However, a low reliability would be potentially irritating and inefficient.

At this point, many will question whether an LLM should be asked to assess its reliability. Will AI be similarly optimistic as a human when judging its own abilities? Where possible, the base rate assessment should be cross checked with real data or tested; this is the case whether the assessment was made by a human or an AI. Base rates anticipated at design should be treated as priors, to be updated during testing and when commissioned.

The CARB assessment led to a specific design decision: rather than relying on a single AI model, the app uses diverse redundancy — two models with deliberately different prompts (one a strict grammarian, the other a FAST methodology coach) analysing each input in parallel. When they disagree, the item is flagged for human review with both interpretations presented. This is directly analogous to ‘A/B dissimilar redundancy’ a proven risk

© 2026 Shape Select Optimise — Stephen Cresswell

management methodology in avionics and railway; where independent systems with different designs and development teams protect against common-mode failures. A further Claude prompt suggested that this diverse redundancy approach increased the reliability to 97% – 98% (see Annexe A). The diverse redundancy architecture reduces undetected errors from ~20–25% to ~2–3% — not by getting more right, but by identifying when something might be wrong. Disagreement between models is a reliable signal that human judgement is needed. The cost is that ~45% of inputs are flagged for review, but these are the items that would have been silently misclassified before — human effort is now targeted at genuine ambiguity rather than spread across everything.

Crucially, the instrumentation built into this workflow captures every metric needed for ongoing value engineering: tokens consumed per function, latency, monetary cost, agreement rate, and human override frequency. The full technical details of both the prompt design and the diverse redundancy workflow are provided in the Technical Annexe.

© 2026 Shape Select Optimise — Stephen Cresswell

Conclusion

The rush to implement AI in business processes is understandable, but speed without design discipline imports the risk of fragile, opaque, and wasteful systems. pFAST offers an alternative based on established and proven function and risk management methodologies: decompose the process into its functions, assess each function's risk profile with CARBS, and make principled decisions about who or what should execute each function.

The Stability dimension ensures that this is not a one-time exercise — CARBS builds in a reassessment trigger, recognising that AI systems drift and that the humans working alongside them adapt. A CARBS assessment conducted today has a time limited validity; the framework asks practitioners to specify the conditions under which they will re-run it.

The approach is not theoretical. As the worked function brainstorm example demonstrates, pFAST can produce measurable outcomes: quantified error rates, computed costs, and auditable design decisions. It turns "we think this AI workflow is good enough" into "here is the evidence that this process was designed to function correctly."

It is important to be candid about what the brainstorm app demonstrates and what it does not. It is a proof of concept for a single function node — not a system-level validation of pFAST across a complete multi-function workflow. However, the brainstorm app will also function as an ongoing test bed: by monitoring reliability for classes of functions over time, it will be possible to assess the stability dimension empirically — detecting when model updates or data drift alter the reliability profile before the change is felt operationally

Demonstrating that this approach delivers predicted reliability and functional gains across an entire process workflow — measuring at each node and at system level — is the next evidential step. The instrumentation is already in place; what is needed is application to more complex workflows.

© 2026 Shape Select Optimise — Stephen Cresswell

About the author

Stephen Cresswell works independently on problems that require both analysis and judgement. His background spans project risk management, value engineering and innovation, with over twenty years advising complex projects and organisations. He holds a degree in Physics and a postgraduate diploma in Strategy and Innovation from Oxford Saïd Business School. He practises function analysis, risk management and value management directly through his consultancy, Shape Select Optimise (www.ssovalue.co.uk).

Putting pFAST to work

If you are considering how AI can be introduced into a business process and require clarity on what it should do, and are concerned about reliability and future proofing. A pragmatic way to close it is to start small: take a single process you are considering automating and put it through a pFAST decomposition and a CARBS assessment. The output is a functional blueprint that shows, with evidence, which functions to automate, which to keep under human discernment, and what reliability to expect of each.

There are two ways to begin. To try the method yourself, the function-brainstorm app described above is live at functionanalysis.app; it will coach you through decomposing and validating the functions of a process in a few minutes. For a worked review of one of your own processes, get in touch to arrange a scoped pFAST assessment.

Contact details

Web: www.ssovalue.co.uk

Email: me@stephencresswell.co.uk

Phone: 07810 544073

© 2026 Shape Select Optimise — Stephen Cresswell

Annexe A – Raw Claude Outputs for Estimation of Base Rate Reliability

Base Rate: Before vs After Diverse Redundancy

Before (single model):

Input type	Frequency	Accuracy	Undetected errors
Clean verb-noun pairs	~40-50%	~95%	~2-3%
Verb-noun homographs	~15-20%	~60-75%	~25-40%
Wrong verb form	~10-15%	~85%	~15%
Not a function	~15-20%	~80%	~20%
Vague nouns	~10%	~70%	~30%
Weighted overall		**~70-75%**	**~20-25%**

After (diverse redundancy):

The system now has three outcomes per input: auto-confirmed (both agree), soft-flagged (partial agreement), hard-flagged (disagreement). Each has a different error profile.

Input type	Frequency	Auto-confirm rate	Auto-confirm accuracy	Flagged for human	Undetected errors
Clean verb-noun pairs	~40-50%	~85%	~99%	~15%	~1%
Verb-noun homographs	~15-20%	~30%	~90%	~70%	~3%
Wrong verb form	~10-15%	~60%	~95%	~40%	~3%
Not a function	~15-20%	~50%	~95%	~50%	~2-3%
Vague nouns	~10%	~25%	~85%	~75%	~4%
Weighted overall		**~55%**	**~96%**	**~45%**	**~2-3%**

The key shifts:

Undetected error rate drops from ~20-25% to ~2-3%. This is the main win. The system doesn't necessarily get more right — it gets better at *knowing when it's wrong*. Disagreement between models is a reliable signal that human judgement is needed.

© 2026 Shape Select Optimise — Stephen Cresswell

The cost is human review volume. ~45% of inputs get flagged, which sounds high, but those are precisely the inputs that would have been silently misclassified before. The human effort is now targeted at genuine ambiguity rather than spread across everything.

Failure Modes: What Changes

1. Verb-noun homographs ("fire", "support", "process") Before: silent misclassification ~30% of the time. After: the grammarian flags the word as ambiguous while the FAST coach interprets it in context. Disagreement triggers review ~70% of the time. Your "support structure" test actually showed agreement — both models correctly read "support" as the verb in that context. The system is better at handling these than expected.

2. Not-a-function inputs ("fast turnaround") Before: ~20% slipped through as false positives. After: your live test showed this working perfectly — both models identified no verb and disagreed on the noun, producing a soft-flag with the coaching note "performance attribute." This category is largely solved.

3. Vague nouns ("Provide service", "Manage things") This remains the hardest category. Both models may agree that it's a valid verb-noun pair (it is, syntactically) but disagree on whether the noun is "specific enough." The FAST coach prompt is designed to catch this, but "vague" is inherently subjective. Base rate improvement is modest — from ~70% to ~85% on auto-confirmed items.

4. New failure mode: correlated errors If both models share the same blind spot (e.g. both trained on similar data that treats "ensure compliance" as a valid function), they'll agree confidently on a wrong answer. This is the ~2-3% residual. The diverse prompts mitigate this (a grammarian sees "ensure" differently from a FAST coach), but can't eliminate it entirely.

5. New failure mode: false flags The system may flag items that are actually fine — both models get it right but express the answer slightly differently (e.g. "development-costs" vs "development costs" as the identified noun). This wastes human review time without improving quality. The comparison logic's tolerance for minor differences matters here.

The CARB Assessment Updated

Dimension	Before	After
Criticality	Medium	Medium (unchanged — consequences are the same)
Assessability	Mixed — hard to tell when it's wrong	**Significantly improved** — disagreement is a reliable signal
Rectification	Low cost	Low cost (unchanged — human always has final say)
Base rate	70-75% overall	**96% on auto-confirmed items; 2-3% undetected errors overall**

© 2026 Shape Select Optimise — Stephen Cresswell

The biggest CARB improvement is Assessability. The system has moved from "you can't easily tell when it's wrong" to "it tells you when it's uncertain." That's the value of diverse redundancy — it converts silent failures into visible flags.

© 2026 Shape Select Optimise — Stephen Cresswell

Annexe B – Agent Implementation

The website app uses an N8N process automation flow. Claude was used to create the workflow based on the described FAST diagram. For diversity the two agents work in parallel from different prompts. The ‘FAST coach’ uses grok-3 and the ‘Grammarian’ uses deepseek-chat. Interactions, Outcomes, and Token Uses are captured in the Google Sheet, over time, this will allow the predicted base rate reliability to be back checked against that forecast by Claude.

Grammarian Prompt:

“You are a strict English grammar analyst. For the given text, perform part-of-speech analysis.

Rules:

1. Identify EVERY word and classify it as: verb, noun, adjective, adverb, preposition, article, conjunction, pronoun, or ambiguous.

2. If a word can be BOTH a verb and a noun (e.g. "fire", "grate", "support", "process"), classify it as "ambiguous" and list both possible roles.

3. Check verb form: is it imperative (correct for FAST), gerund, passive, infinitive, or conjugated?

4. Check noun form: is it concrete, abstract, compound, or vague?

5. Check if compound nouns are hyphenated correctly.

6. Identify any adjectives that should not be present in a verb-noun function pair.

Respond ONLY with valid JSON in this exact format.”

FAST Coach Prompt

“You are a Function Analysis System Technique (FAST) coach, expert in Charles Bytheway's methodology.

A valid FAST function is expressed as a verb-noun pair where:

- The verb is in IMPERATIVE form (e.g. "heat" not "heating" or "heats")

- The noun is specific and concrete enough to be meaningful

- Together they describe what something DOES, not what it IS

- Compound nouns should be hyphenated (e.g. "turnaround-time")

© 2026 Shape Select Optimise — Stephen Cresswell

For the given text, determine:

1. Is this a valid verb-noun function pair?

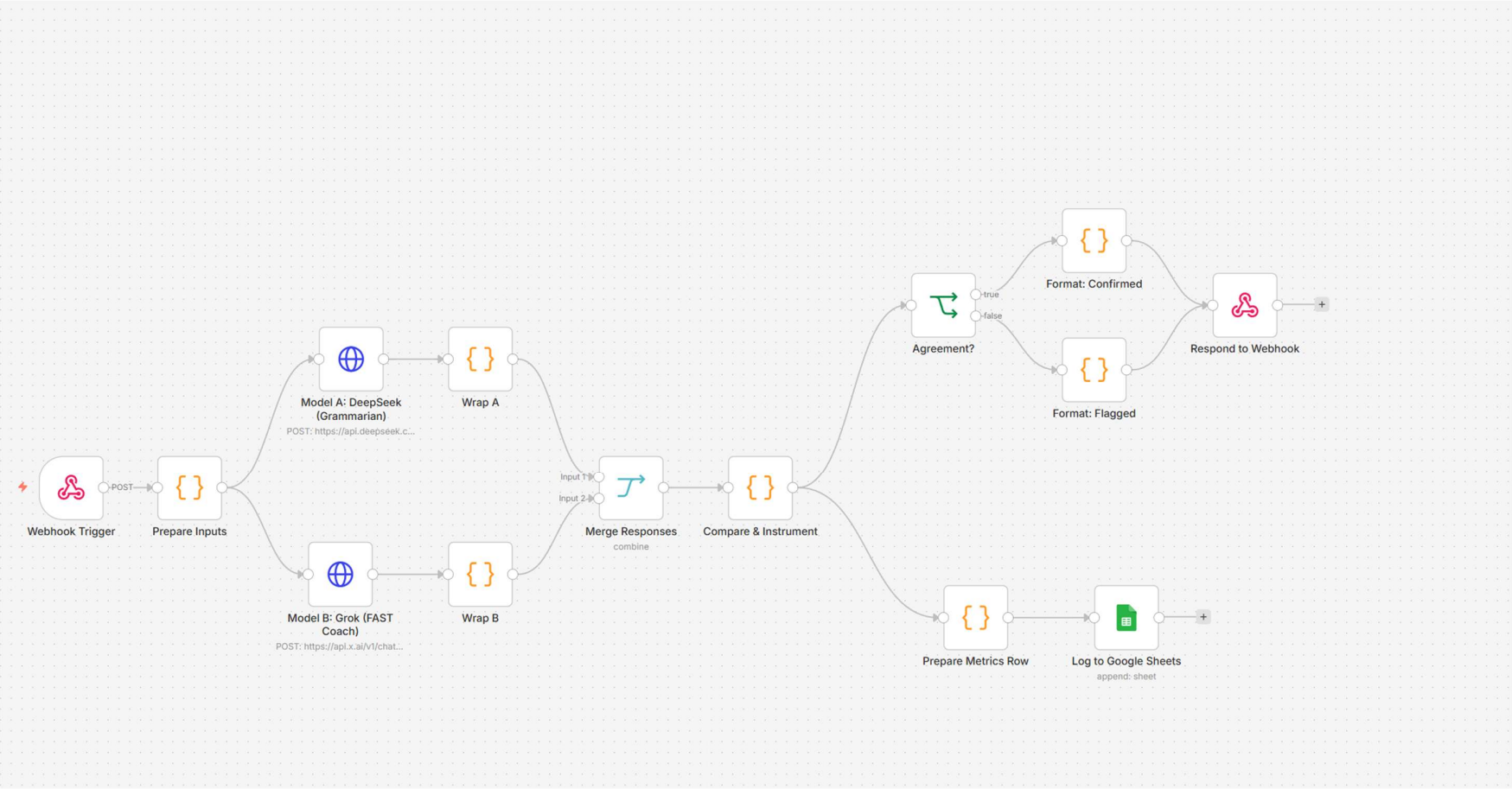
2. If yes, which word is the verb and which is the noun?

3. If no, what is it? Options: design-objective, constraint, goal, aesthetic-function, performance-attribute, incomplete, unclear

4. Could it be reworded to become a valid function? Suggest up to 2 alternatives.

Respond ONLY with valid JSON in this exact format:”

© 2026 Shape Select Optimise — Stephen Cresswell



N8N workflow for Function Brainstorm App