



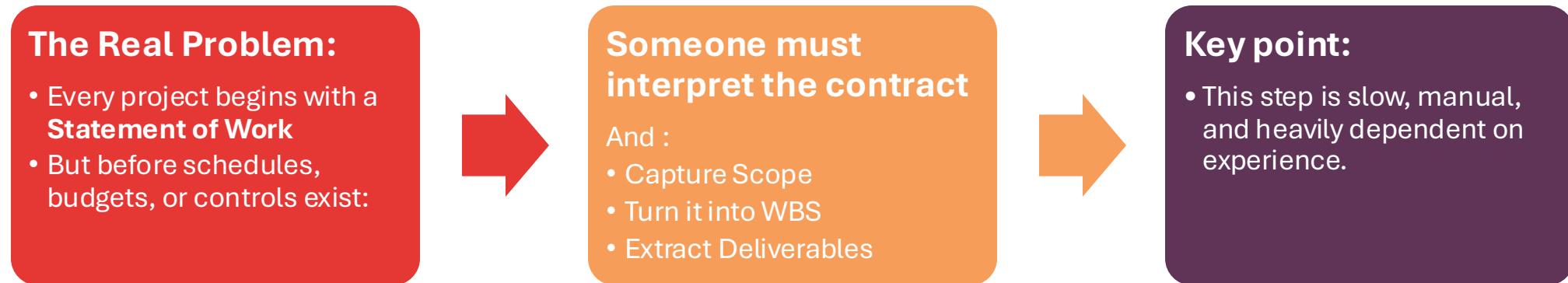
Scaling Project Controls: When AI Learns WBS Thinking

Presented by:
Daniel Foster & Az Pathan



Project Bee
UTILIZING AI FOR PROJECT CONTROLS

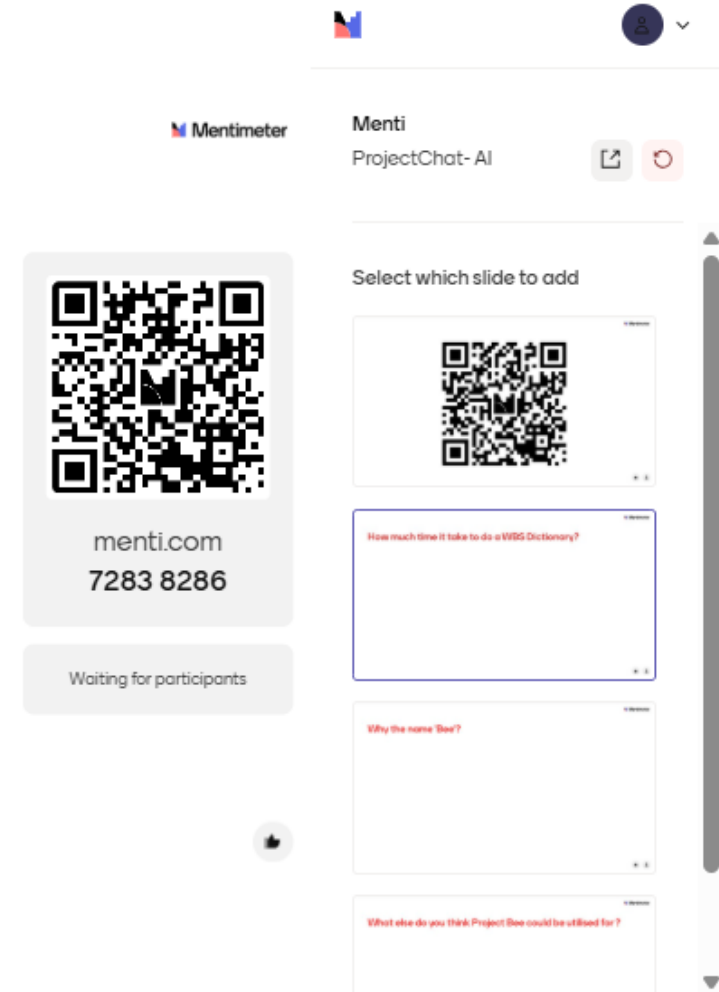
A Hidden Bottleneck



How much time it take to do a WBS Dictionary?

How much time it take to do a WBS Dictionary?

Join at menti.com | use code 7283 8286



The screenshot shows a Menti poll interface. At the top, there is a Menti logo and a user profile icon. Below that, the poll title "Menti ProjectChat- AI" is displayed with share and refresh icons. The main content area is titled "Select which slide to add" and contains three slide thumbnails. The first thumbnail shows a QR code. The second thumbnail shows the question "How much time it take to do a WBS Dictionary?". The third thumbnail shows the question "Why the name 'Bee'?". The fourth thumbnail shows the question "What else do you think Project Bee could be utilized for?". A vertical scrollbar is on the right side of the slide selection area. At the bottom of the interface, there is a thumbs-up icon.

Why Skeiny Is Building Project Bee



We're building AI to help project teams do more with the capacity they have—reducing manual effort and helping focus on high-value decision making in complex projects.

Project Controls Don't Scale

Complex contracts

- Hundreds of clauses
- Ambiguous responsibilities



Reality:

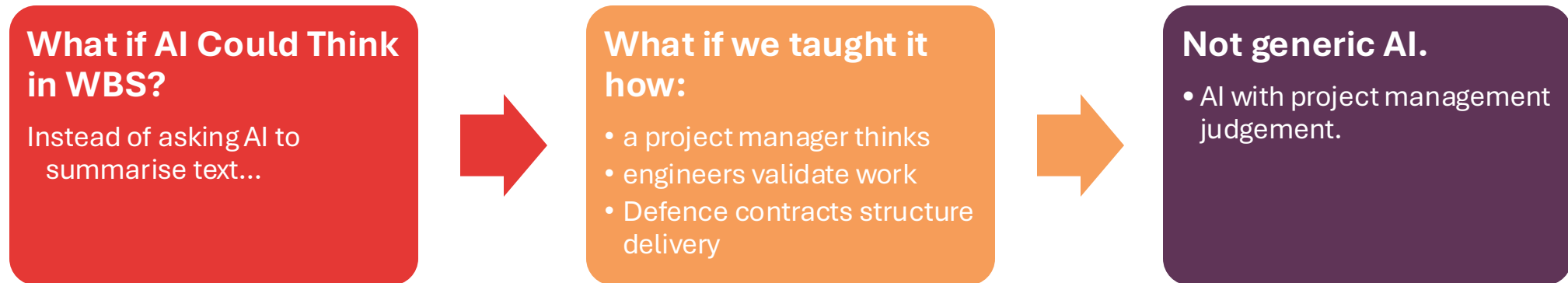
- Projects are becoming more complex.
- Teams are not getting bigger.
- We need ways to amplify the capability of project teams
- Senior PMs can do it well. But organisations cannot scale senior PM thinking.
- AI is transforming many industries.



Result:

- Weeks spent building WBS
- Inconsistent interpretations
- Missed scope.

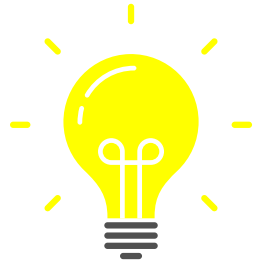
The Moment of Insight



Introducing



Project Bee
UTILIZING AI FOR PROJECT CONTROLS



Initial Goal was to use LLM to create WBSD with mapping of each SOW clause

Project Bee Utilizes

- LLM to wear Project Control Professional's hat



Project Bee explores how to:

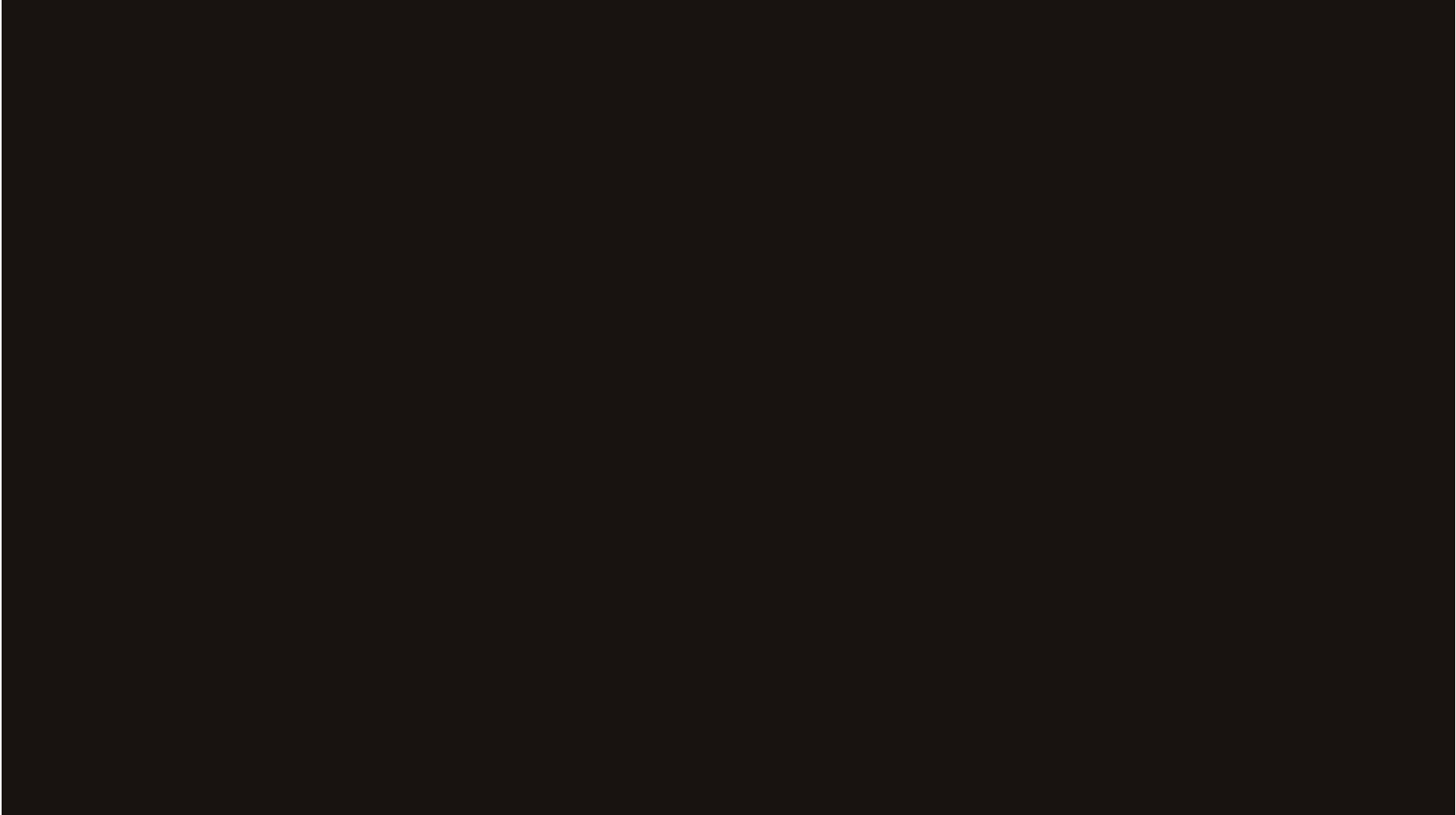
- Parse SOW clauses
- Interpret intent
- Map them into WBS dictionaries



Through:

- Prompt engineering

Why the name 'Bee'



Why “Bee”?

Why Project Bee?

Bees create highly structured systems through collective intelligence.

Project Bee uses:

- AI
- project controls knowledge
- engineering expertise

to build efficient and structured project controls frameworks.



The First Attempt (Reality Check)

Our First Attempt Failed Initial approach:

- Prompt engineering
- Hard-coded prompts
- Basic automation



Result:

- fragile
- inaccurate
- brittle when contracts changed



Lesson:

- AI alone is not enough.
- need to give better structure and domain knowledge to LLM.



The Breakthrough

The Architecture That Changed Everything

- Give LLM context by use of
 - Vector Memory
 - RAG (Retrieval-Augmented Generation)
- Use Human validation (Human In Loop)



Key technologies:

- n8n (workflow orchestration)
- Supabase (vector knowledge base)

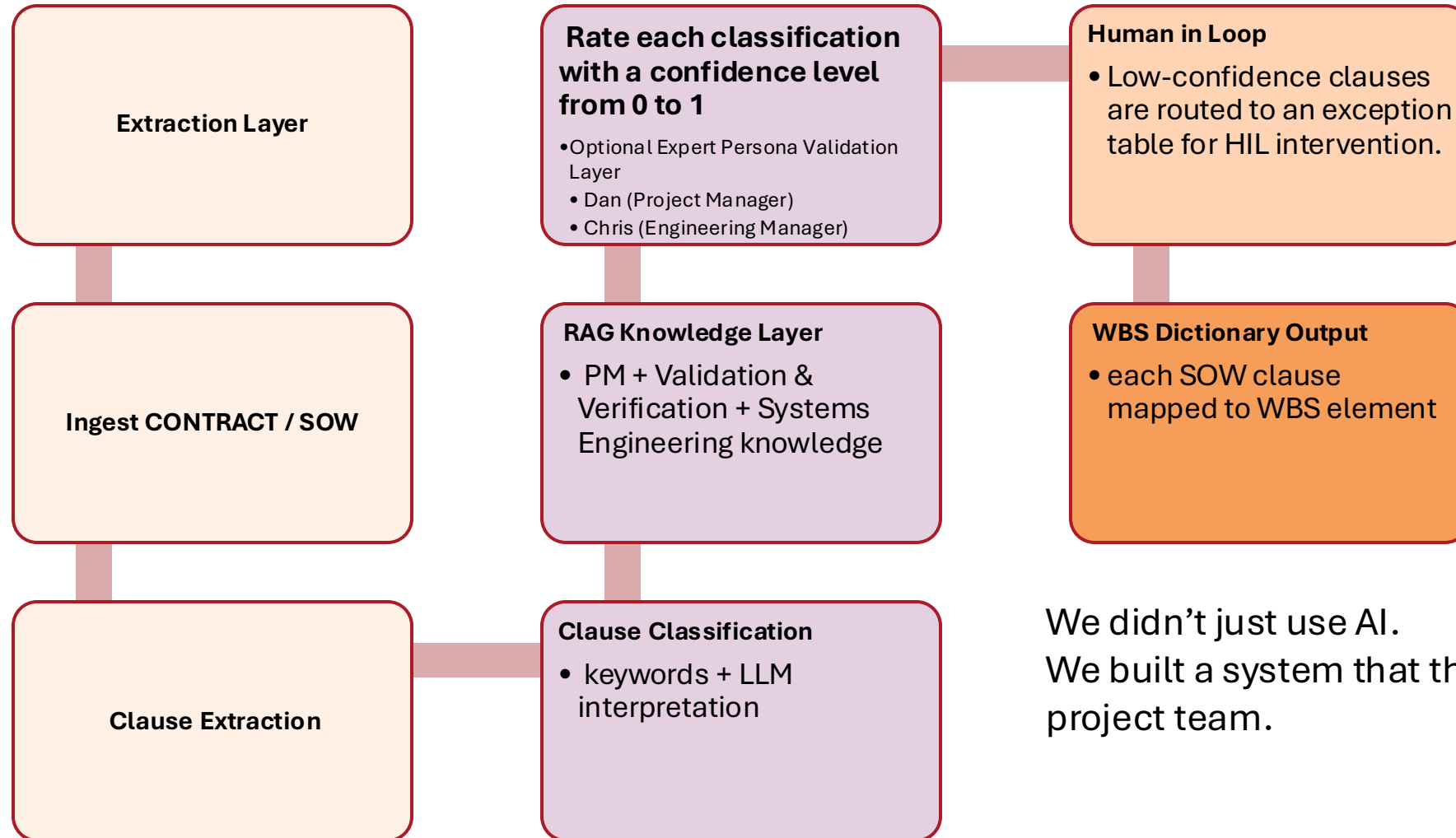


Key insight:

- The model needed context, not just prompts.



The Architecture



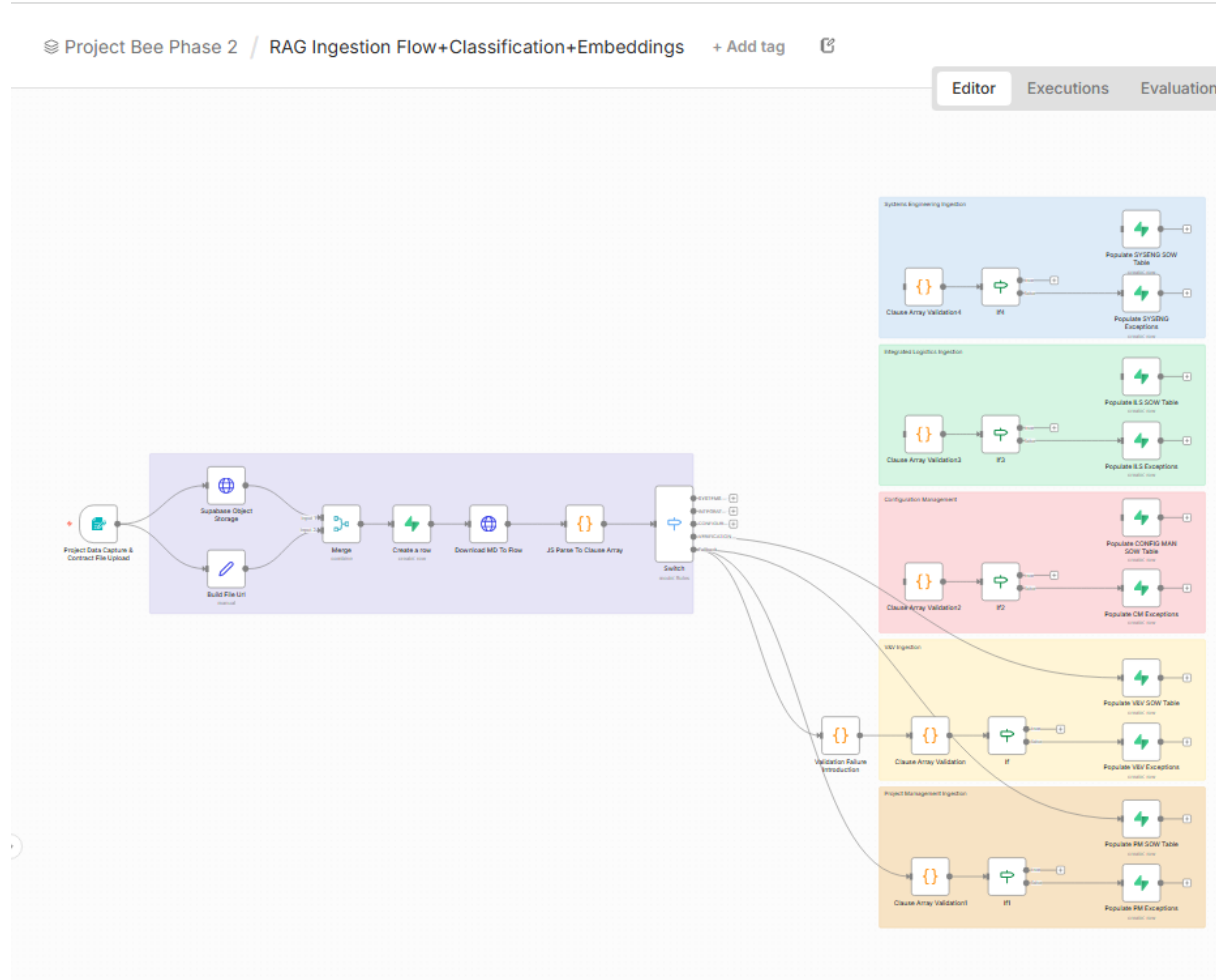
We didn't just use AI.
We built a system that thinks like a project team.

Teaching AI How PMs Think

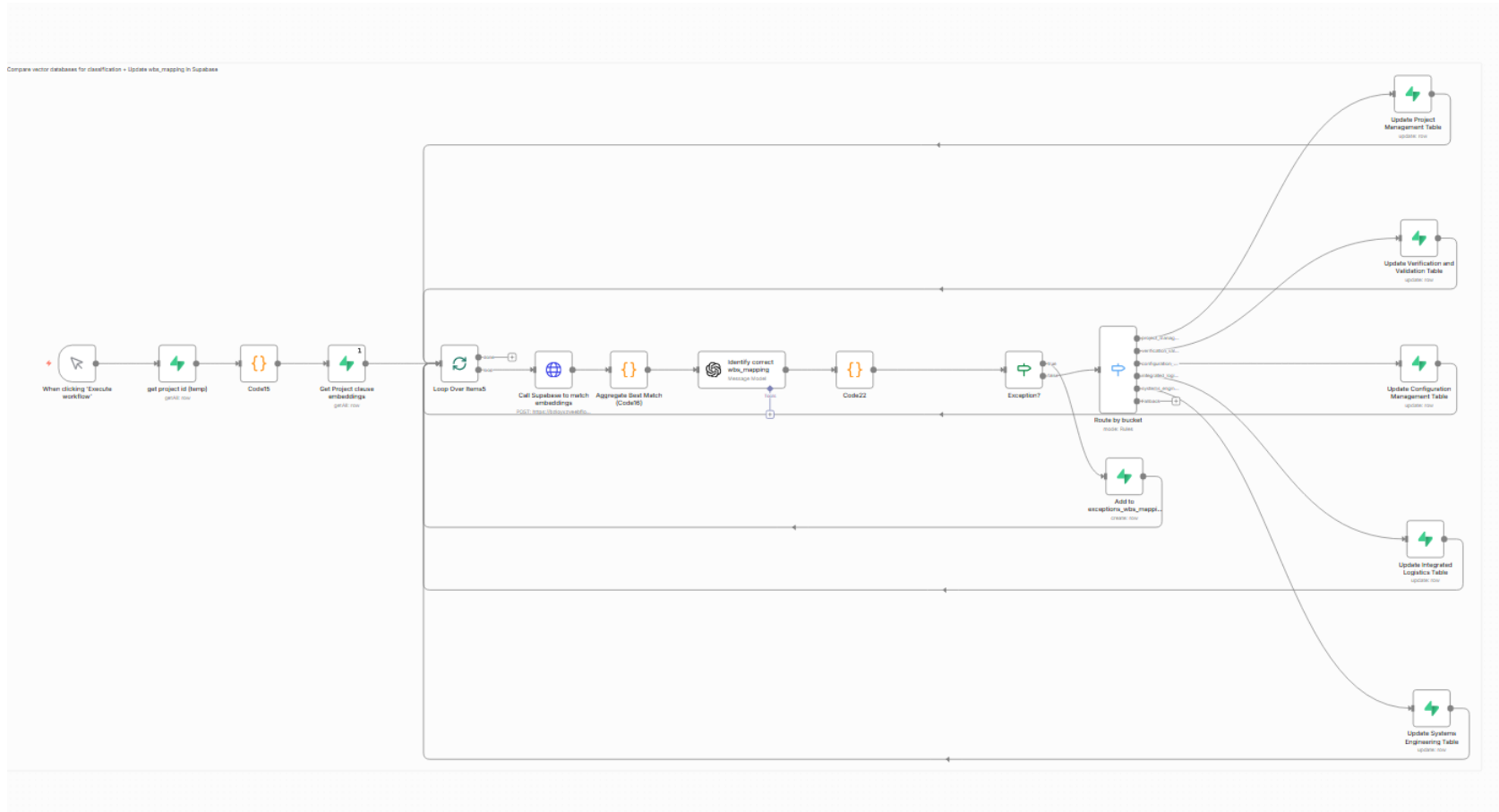


** System initially built to classify SOW clauses for V&V and PM with standard Waterfall WBS for acquisition project using standard AUSDEF contract SOW and template*

The Ingestion Architecture



The Reasoning Layer Architecture



Project Bee in Action: Demo

Ingestion and Embedding

This screenshot shows a workflow titled 'Final_RAG_Ingestion' with 0/2 active steps. The workflow starts with 'Project Data Capture & Contract File Upload', which branches into 'Supabase Object Storage' and 'Build File Uri'. Both lead to a 'Merge' step. This is followed by 'Create a row', 'Download MD To Flow', and 'JS Parse To Clause Array'. The final step is a 'Switch' node with rules for HYPERTEXT, INTEGRATE, CONFIGUR, and VERIFICATION. The interface includes a sidebar with 'Clause Array', 'Integrated Log', and 'Configuration' sections, and a red 'Execute workflow' button at the bottom.

Reasoning and Categorization



This screenshot shows a workflow titled 'Final_RAG_Reasoning' with 0/3 active steps. The workflow is a complex sequence of steps including 'When using source workflow', 'get project id from contract', 'CALL', 'Get Project clause embeddings', 'Loop Over Rows', 'Call Embedder to create embeddings', 'Aggregate Row Match Scores', 'Map', 'Identify relevant rows, prepare', 'Embed', 'Embedder', 'Build to vector', and 'Update Configuration Management Data'. The interface includes a sidebar with 'Clause Array' and 'VBI Ingestion' sections, and a red 'Execute workflow' button at the bottom.

Live Example

Clause 7.1.3: Verification Cross Reference Matrix

- ****7.1.3.1**** The Contractor shall develop, deliver and update a Verification Cross Reference Matrix (VCRM) in accordance with CDRL Line Number V&V-200, which captures the detail of the Verification criteria and status of the Verification activities for each requirement in:



Then maps to:

- Verification & Validation
- With Confidence level of 97% and
- with a reasoning
 - *The clause is about producing a Verification Cross Reference Matrix tied to a V&V CDRL deliverable (V&V-200). All top-scoring candidates map to WBS 1.03.10, making it the correct alignment.*



Key point:

- This is structured reasoning, not guessing.

The Output

← / Skeiny FREE / ProjectBe

Table Editor

schema public

+ New table

Search tables...

- exceptions_configur... UNRESTR
- exceptions_integrat... UNRESTRIC
- exceptions_project... UNRESTRIC
- exceptions_systems... UNRESTRIC
- exceptions_verificati... UNRESTRIC
- exceptions_wbs_... UNRESTRIC
- master_project_... UNRESTRIC
- project_clause_em... UNRESTRIC
- sow_configuration_... UNRESTRIC
- sow_integrated_l... UNRESTRIC
- sow_project_man... UNRESTRIC
- sow_systems_engi... UNRESTRIC
- sow_verificatio... UNRESTR
- wbs_embeddings UNRESTRICTED
- wbs_keywords
- wbs_library



Output with ~80–90% accuracy for initial clause classification, with low-confidence routed for human-in-the-loop validation.

	body text	raw_md text	wbs_map text	created_at timestamptz	project_id text	llm_confidence float8	llm_reason text
<input type="checkbox"/>	EMPTY	**7.1** **V\&V Managen	1.03.01	2026-01-25 22:12:39.495187+00	dba0a182-e179-46de-8745-637c02977cd9	0.98	Direct alignment between cl
<input type="checkbox"/>	EMPTY	**7.1.1** **General**	1.03.01	2026-01-25 22:12:39.495187+00	dba0a182-e179-46de-8745-637c02977cd9	0.92	The clause's generic nature f
<input type="checkbox"/>	The Contractor acknowledges that the ob	**7.1.1.1** The Contracto	1.03.01	2026-01-25 22:12:39.495187+00	dba0a182-e179-46de-8745-637c02977cd9	0.92	The clause speaks to overarc
<input type="checkbox"/>	integrate the V\&V of the Mission System	**7.1.1.1.a** a. integrate ti	1.03.01	2026-01-25 22:12:39.495187+00	dba0a182-e179-46de-8745-637c02977cd9	0.92	1.03.01 explicitly covers V&V
<input type="checkbox"/>	establish and maintain a strategy of progr	**7.1.1.1.b** b. establish a	1.03.01	2026-01-25 22:12:39.495187+00	dba0a182-e179-46de-8745-637c02977cd9	0.91	The clause describes overarc
<input type="checkbox"/>	EMPTY	**7.1.2** **V\&V Plannin	1.03.01	2026-01-25 22:12:39.495187+00	dba0a182-e179-46de-8745-637c02977cd9	0.88	V&V Planning fits squarely ur
<input type="checkbox"/>	The Contractor shall develop, deliver and	**7.1.2.1** The Contractc	1.03.10	2026-01-25 22:12:39.495187+00	dba0a182-e179-46de-8745-637c02977cd9	0.97	Exact match to V&V-100 CDF
<input type="checkbox"/>	The Contractor shall conduct all V\&V act	**7.1.2.2** The Contract	1.03.01	2026-01-25 22:12:39.495187+00	dba0a182-e179-46de-8745-637c02977cd9	0.92	The clause directly addresser
<input type="checkbox"/>	EMPTY	**7.1.3** **Verification C	1.03.10	2026-01-25 22:12:39.495187+00	dba0a182-e179-46de-8745-637c02977cd9	0.86	The VCRM is a standard verifi
<input type="checkbox"/>	The Contractor shall develop, deliver and	**7.1.3.1** The Contractc	1.03.10	2026-01-25 22:12:39.495187+00	dba0a182-e179-46de-8745-637c02977cd9	0.97	The clause is about produc
<input type="checkbox"/>	the FBL for the Mission System; and	**7.1.3.1.a** a. the FBL fo	1.03.01	2026-01-25 22:12:39.495187+00	dba0a182-e179-46de-8745-637c02977cd9	0.63	FBL establishment/validation
<input type="checkbox"/>	the FBL for the Support System.	**7.1.3.1.b** b. the FBL fo	1.03.01	2026-01-25 22:12:39.495187+00	dba0a182-e179-46de-8745-637c02977cd9	0.63	The clause aligns with establ
<input type="checkbox"/>	The Contractor shall provide all facilities	**7.1.3.2** The Contract	1.03.01	2026-01-25 22:12:39.495187+00	dba0a182-e179-46de-8745-637c02977cd9	0.82	VCRM access is part of V&V
<input type="checkbox"/>	EMPTY	**7.1.4** **Previous V\&	1.03.10	2026-01-25 22:12:39.495187+00	dba0a182-e179-46de-8745-637c02977cd9	0.72	The clause focuses on supply
<input type="checkbox"/>	The Contractor shall develop, deliver and	**7.1.4.1** The Contractc	1.03.10	2026-01-25 22:12:39.495187+00	dba0a182-e179-46de-8745-637c02977cd9	0.97	Direct keyword match (V&V-

What This Enables



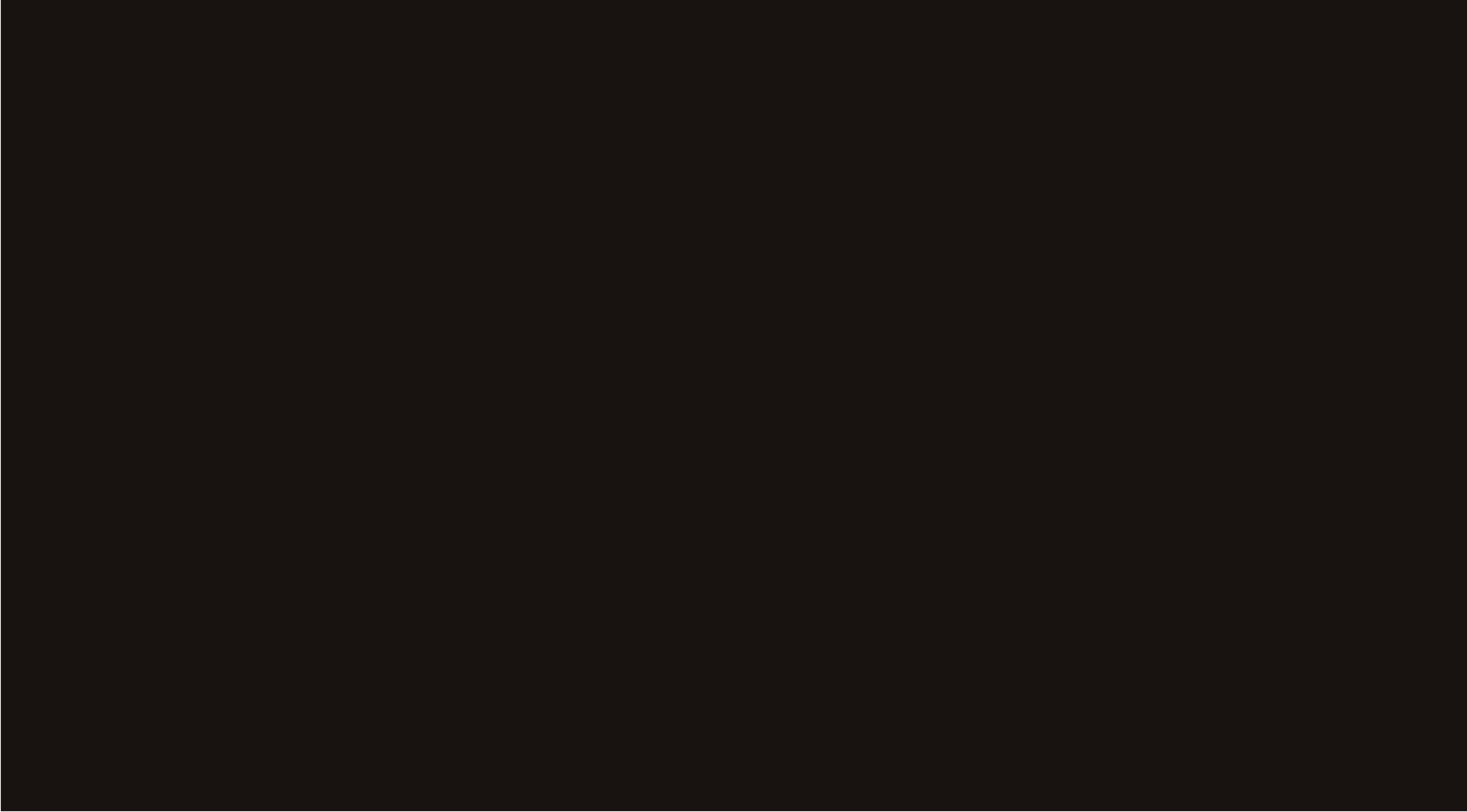
The Bigger Opportunity



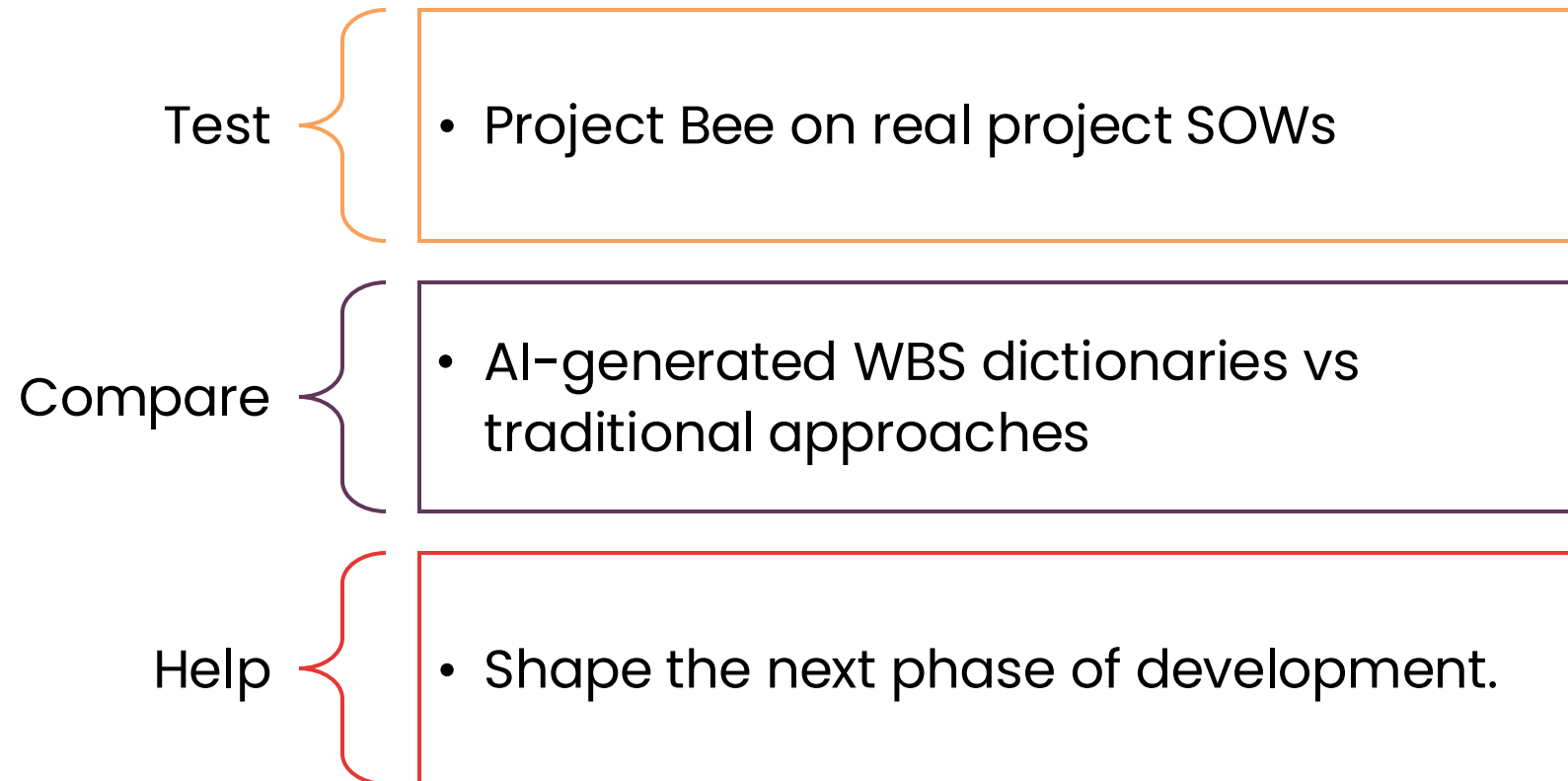
Today Project Bee helps generate a WBS Dictionary from the SOW.

Tomorrow it could help generate the risk register, compliance matrix, and even the first draft of the schedule logic

What else do you think Project Bee could be utilised for ?



We Are Open to Collaboration Opportunities



Questions ?



Thankyou

Daniel Foster

Managing Principal Partner

P 0406 403 816

E dan@skeiny.com.au

This presentation contains Skeiny Intellectual Property and must not be shared outside of the intended organisation.

