CODESPELL

Case Study

From **Scattered Inputs to Structured Success** with CodeSpell

About the Customer

The customer is a major player in the North American transportation and logistics industry, delivering comprehensive freight, specialized, and final-mile services through a large nationwide network.

The Challenge and the Need

Across their software development lifecycle, the customer faced inefficiencies and risks:

- Requirements were scattered across unstructured sources, slowing consensus and traceability.
- Backend scaffolding was repetitive and consumed excessive time.
- UI validation patterns were inconsistent, leading to maintenance complexity and errors.
- Large-scale refactorinag and change tracking posed significant logistical challenges.

The Solution

1. Centralizing Requirements with ReqSpell

Implementation of ReqSpell provided a structured, queryable repository for requirements, with use cases such as:

- Gathering screen details, including purpose, user stories, acceptance criteria, navigation, and technical notes.
- Capturing process flows and business rules like input handling, error management, and navigation logic.
- Documenting data validations-input context, security, error rules.
- Mapping abbreviations embedded in UI code and field definitions.
- Retrieving sub-program and modular function details for easier maintenance.

Impact:

- Accuracy: ~40%, except sub-program details at ~20%
- Efficiency: ~50%, except sub-program details at ~20%
- Time saved: ~20–50%, varying with complexity.

2. Accelerating Backend Code Generation & Refactoring

CodeSpell enabled substantial productivity gains in backend workflows:

- **DB-Design-based code generation** reduced effort from **8 to 3 hours**, saving 5 hours per cycle (62.5% reduction, velocity up 266%).
- **Project-wide generation** ensured consistency in a single pass.
- **Validator refactoring** cut effort from 2–4 hours to 1–3 hours (25–50% reduction, velocity up 33–200%).
- **Workspace-level impact analysis** enabled efficient multi-file updates like schema changes..

3. Optimizing UI Development & Validation Consistency

The front-end workflows realized major efficiency gains:

- **Validation pattern reuse** dropped effort from 4–16 hours to 2–6 hours (50–62.5% reduction, velocity up 200–266%).
- **Prompt engineering for UI & error handling** slashed effort from 2–4 hours to 30–60 minutes (75% reduction, velocity up 400%).
- **UI refactoring** turned large, unwieldy HTML code into modular, maintainable components.
- **Integration of external services** saw effort reduced from 4–16 hours to 2–8 hours (50% reduction, velocity up 200%). updates like schema changes.

Results & Impact

Use Case	Time Before	Time with CodeSpell	Saved	Efficiency	Velocity Gain
ReqSpell	Manual effort	~50–80% of manual	20–50%	~50% (20% sub-program)	Faster onboarding
DB Code Generation	8 hrs	3 hrs	5 hrs	62.5%	266%
Validation Pattern Reuse	4–16 hrs	2–6 hrs	2–10 hrs	50–62.5%	200–266%
Prompt Engineering (UI)	2–4 hrs	0.5–1 hr	1.5–3 hrs	75%	400%
Validator Refactoring	2–4 hrs	1–3 hrs	1 hr	25–50%	33–200%
External Service Integration	4–16 hrs	2–8 hrs	2–8 hrs	50%	200%

Key Benefits Realized

- Centralized requirement documentation and traceability.
- Scalable backend scaffolding, with consistent and repeatable outputs.
- Faster UI development, with modular code and consistentvalidation logic.
- Reduced maintenance risk through pattern reuse and global updates.
- Universal efficiency gains across both backend and frontend workflows.

Looking Ahead

With this momentum, the customer is poised to extend CodeSpell's capabilities across additional domains and applications-scaling delivery speed, consistency, and quality throughout their software ecosystem.

Ready to Improve Your Operations?

CodeSpell is an GenAI SDLC platform that helps enterprise teams automate code generation, testing, and infrastructure setup, seamlessly integrated within your IDE.
For inquiries, reach out to us at Info@codespell.ai

www.codespell.ai
Follow us: