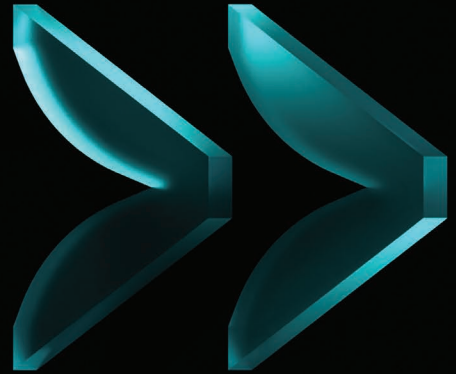


Transforming Oracle Forms Documentation with ReqSpell



A large retailer based in the United States faced significant challenges with legacy Oracle Forms documentation. Each form contained multiple components such as Blocks, Program Units, Triggers, LOVs, and Record Groups, all of which were being manually documented by teams across the organization.

Why Change Was Necessary



Legacy Code

made enhancements and debugging difficult due to an outdated and hard-to-read codebase that lacked modularity or clear documentation.



Complex Oracle Forms

added to the challenge, with each form comprising extensive logic, components, and customizations that increased the learning curve.



High Technical Depth

meant each user story involved navigating multiple triggers, procedures, and data flows, requiring deep architectural understanding.



Simultaneous Execution of User Stories

posed coordination issues, with overlapping dependencies leading to inconsistencies and delayed delivery.



Context Loss in Individual Stories

made it difficult to maintain continuity, as isolated work on user stories led to fragmented understanding and reduced quality.

The ReqSpell Intervention

- Leveraged ReqSpell's intelligent Oracle Forms chunking to break down complex forms into logical units such as programs, blocks, triggers, and nested elements for accurate parsing.
- Used the built-in pre-processor to automatically detect program units and deeply nested structures, improving contextual accuracy for large language model analysis.
- Applied ReqSpell's optimized reverse engineering workflow to handle bulk form processing, enabling simultaneous documentation of multiple user stories without losing context.

Tech Stack

AI Framework



AI Models



Hosting Service



Authentication



Programming Language



Transformation Overview

Category	Traditional Approach	ReqSpell Approach
Input Preparation	Manually written, form-specific	Structured inputs reused across forms
Document Structure	Varied by user	Standardized with pre-defined prompts
Effort Required	Fully manual	Reduced by approximately 50 percent
Time per Operation	10 to 15 hours	Approximately 1 hour
Time per Form	Around 60 hours	Reduced to 7 hours
Accuracy	Prone to interpretation errors	AI-driven consistency
Reusability	Not possible	High reusability across forms
Collaboration	Limited by style differences	Enhanced with predictable structure
Scalability	Low due to manual effort	High scalability with unified workflow
Output Consistency	Inconsistent	Uniform and standardized
Human Effort	Focused on writing	Shifted to review and validation
Overall Efficiency	Low	High, fast, and auditable

Business Impact Delivered

Reduced manual effort in reverse engineering by up to 70 percent, accelerating documentation delivery.

- Delivered consistent, high-quality outputs using standardized prompts and reusable templates.
- Enabled scalability and reuse across forms and projects, minimizing redundant work.
- Improved accuracy and completeness by reducing human error through AI-guided workflows.
- Preserved documentation standards across teams, ensuring long-term knowledge retention.
- Freed up experts to focus on validation and innovation instead of manual drafting.



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: [!\[\]\(e474458956c9a37fbf9586ddb60a7fa1_img.jpg\)](#) [!\[\]\(4d1d3f2547aeece54bb6babd23f4121b_img.jpg\)](#) [!\[\]\(ec45aa71601db5755c5e2662ad427708_img.jpg\)](#)

©2024, Codespell. All rights reserved.