



COMMON EUROPE CONGRESS 2026

14 - 17 June
Lyon, France

The largest conference in Europe
for solutions around IBM Power (IBM i, AIX, Linux) & IBM Storage

common
EUROPE

www.comeur.org

common
FRANCE

LYON | CENTRE DE CONGRÈS
EVENTS | DE LYON

**Welcome to Lyon, France
and the 2026 Common Europe Congress**

**Bienvenue à Lyon, en France,
et au Congrès de Common Europe 2026**

From Coding Assistant to Engineering Team



Agentic Agile RPG Development for IBM i
with Bob and BMAD

Presented by: Lead Software Engineer

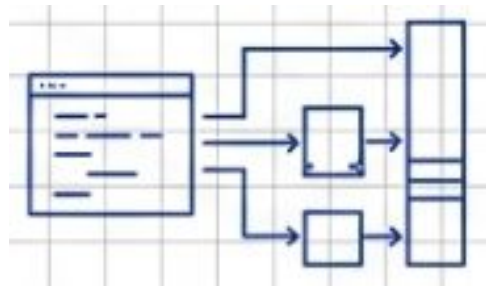
The Evolution of Software Engineering

Shifting the primary bottleneck from hardware to human cognitive bandwidth.

Phase 1

Assembly Language

Direct hardware instruction. Every line of code mapped to a processor operation.



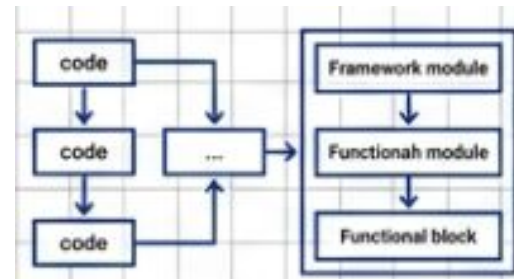
Bottleneck Shift

Execution Speed

Phase 2

High-Level Languages

Algorithmic abstraction and framework utilization. Focus on logical flows.



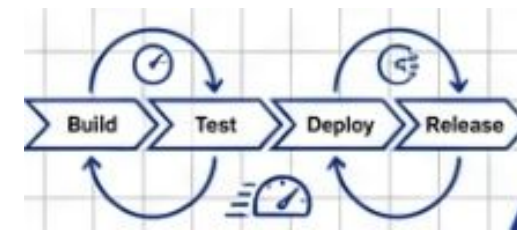
Bottleneck Shift

Development Speed

Phase 3

CI/CD & Automation

Accelerated delivery and deployment cycles. DevOps integration.



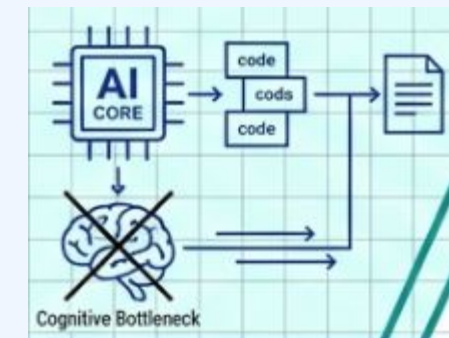
Bottleneck Shift

Deployment Speed

Phase 4 CURRENT ERA

Autonomous Coding

AI-driven generation resolving human cognitive bottlenecks.



The New Frontier

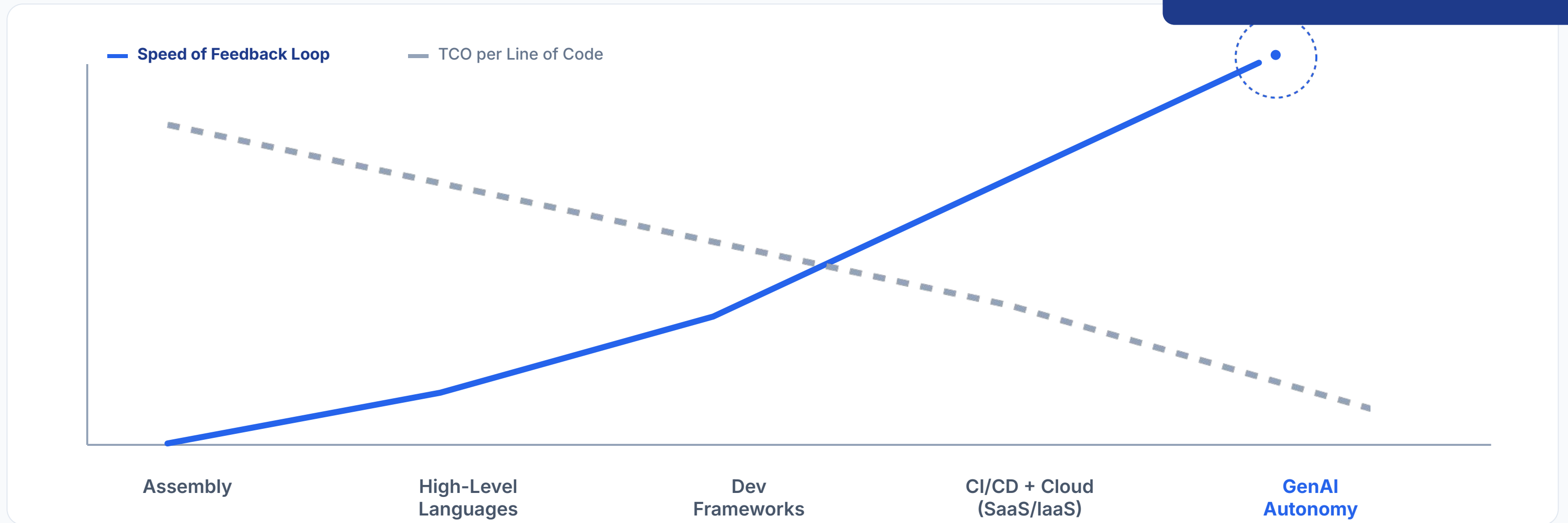
Cognitive Bandwidth

The Era of Automatic Coding

Software Development Evolution & Risk Dynamics

Strategic Impact:

Maximum **feedback speed** and minimal generation cost directly drive project **risk reduction**.



Adoption Tension

Balancing development opportunities with enterprise realities.

Opportunity

- **Developer ⇒ Agent Team Manager**
- **Massive feature output velocity**
- **Accelerated time-to-market**

VS

Enterprise Reality

- **Replicability & Explainability**
- **Compliance & Certification (NIS2, SOC, ISO)**
- **Predictability of Costs**
- **Brownfield modernization (huge legacy codebases)**

Historical standard: "No one ever got fired for choosing IBM."

*Modern standard: "Everyone will be fired if the excuse is: **I don't know, AI did it.**"*

Why a Framework

Development doesn't end when *'it works!'*

The Pillars of Governance

AI Capability & Generative Output


Boundaries (Scope containment)

Rules (Architectural governance)

Compliance & Certification (NIS, SOC, ISO)

Workflows (SDLC phases)

Strategic Outcomes

 Structural constraints automatically generate proper documentation required by auditors.

 Understand the "Why" behind architectural decisions.

 Full SDLC traceability across the entire lifecycle.

Repeating History in the AI Tooling Landscape

Comparing Open vs. Proprietary Ecosystems

Proprietary Ecosystems

Codex, Claude Code, Bob, ...

- **Predictability:** Opaque changes where performance degrades without notice.
- **Privacy & Compliance:** Data transmission to external, black-box providers.
- **Economics:** Rising costs vulnerable to provider pricing alterations.
- **Flexibility:** Locked into proprietary SDLC workflows.

Open Ecosystems

BMAD, DeepSeek, OpenCode, ...

- **Predictability:** Total control over updates and routing.
- **Privacy & Compliance:** Supports true privacy and local Small Language Model (SLM) inference.
- **Economics:** Sustainable cost structures with predictable infrastructure investments.
- **Flexibility:** No vendor lock-in with contributable pathways.

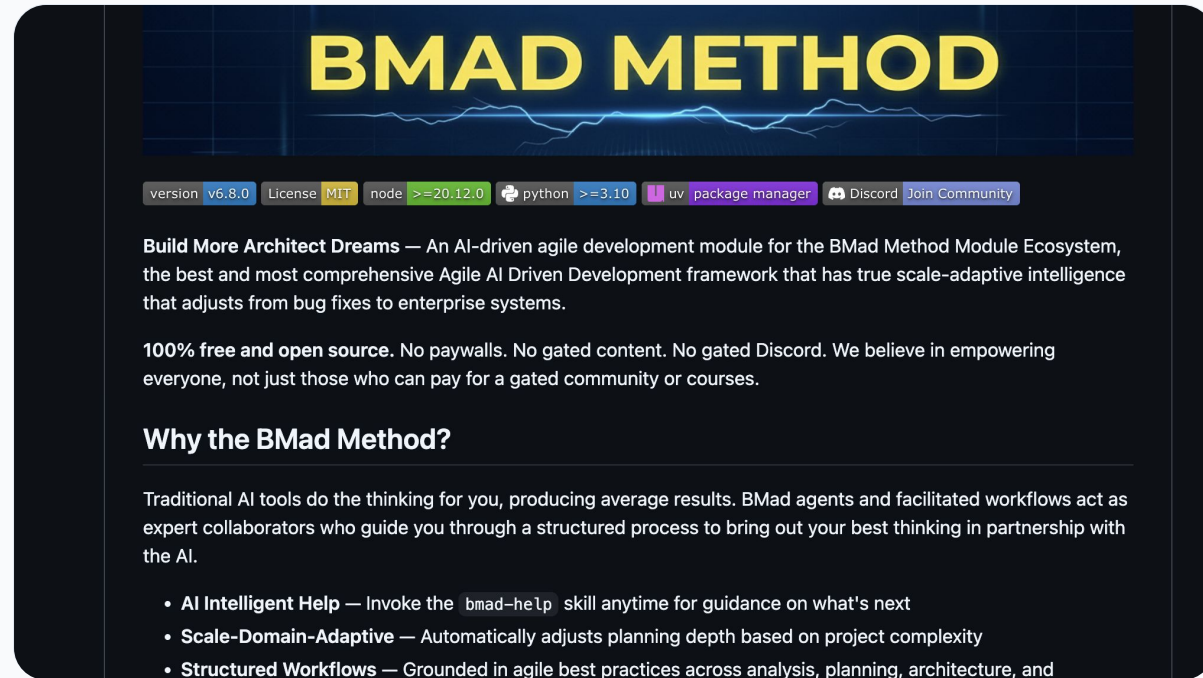
Operational Paradigms

Comparing General Assistants vs. Specialized Agentic Frameworks

Dimension	Bob (General Assistant)	Bob + BMAD (Specialized Agents)
Speed	Immediate / Conversational	Thorough / Phased
Structure	Adaptive / Flexible	Sequential / Methodological
Documentation	As-needed	Comprehensive / Formal
Quality Gates	Optional	Mandatory / Built-in

BMAD Method Ecosystem

An agile development framework for true scale-adaptive intelligence



BMAD METHOD

version v6.8.0 License MIT node >=20.12.0 python >=3.10 uv package manager Discord Join Community

Build More Architect Dreams — An AI-driven agile development module for the BMad Method Module Ecosystem, the best and most comprehensive Agile AI Driven Development framework that has true scale-adaptive intelligence that adjusts from bug fixes to enterprise systems.

100% free and open source. No paywalls. No gated content. No gated Discord. We believe in empowering everyone, not just those who can pay for a gated community or courses.

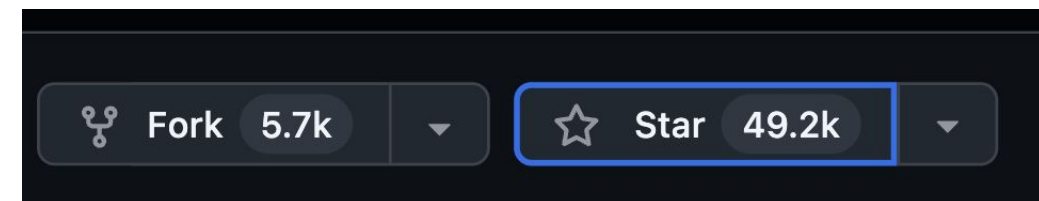
Why the BMad Method?

Traditional AI tools do the thinking for you, producing average results. BMad agents and facilitated workflows act as expert collaborators who guide you through a structured process to bring out your best thinking in partnership with the AI.

- **AI Intelligent Help** — Invoke the `bmad-help` skill anytime for guidance on what's next
- **Scale-Domain-Adaptive** — Automatically adjusts planning depth based on project complexity
- **Structured Workflows** — Grounded in agile best practices across analysis, planning, architecture, and

Open Source Repository

github.com/bmad-code-org



Fork 5.7k Star 49.2k



Why The BMAD Method?



AI Intelligent Help

Invokes the *bead-help* skill anytime for guidance on what's next.



Domain-Adaptive

Automatically adjusts project context based on complexity.



Structured Workflows

Grounded in agile best practices across analysis, planning, and implementation.



Specialized Agents

12+ Domain experts (PM, Architect, Developer) working in unison.

bmad-method install



```
~/works git:(main) $ cd CEC-BMAD/
```

```
~/works/CEC-BMAD git:(main) $ npx bmad-method install
```

```
... installing official and community modules ...
```

bmad-method install

Integrating toolchains and defining identity for agentic development

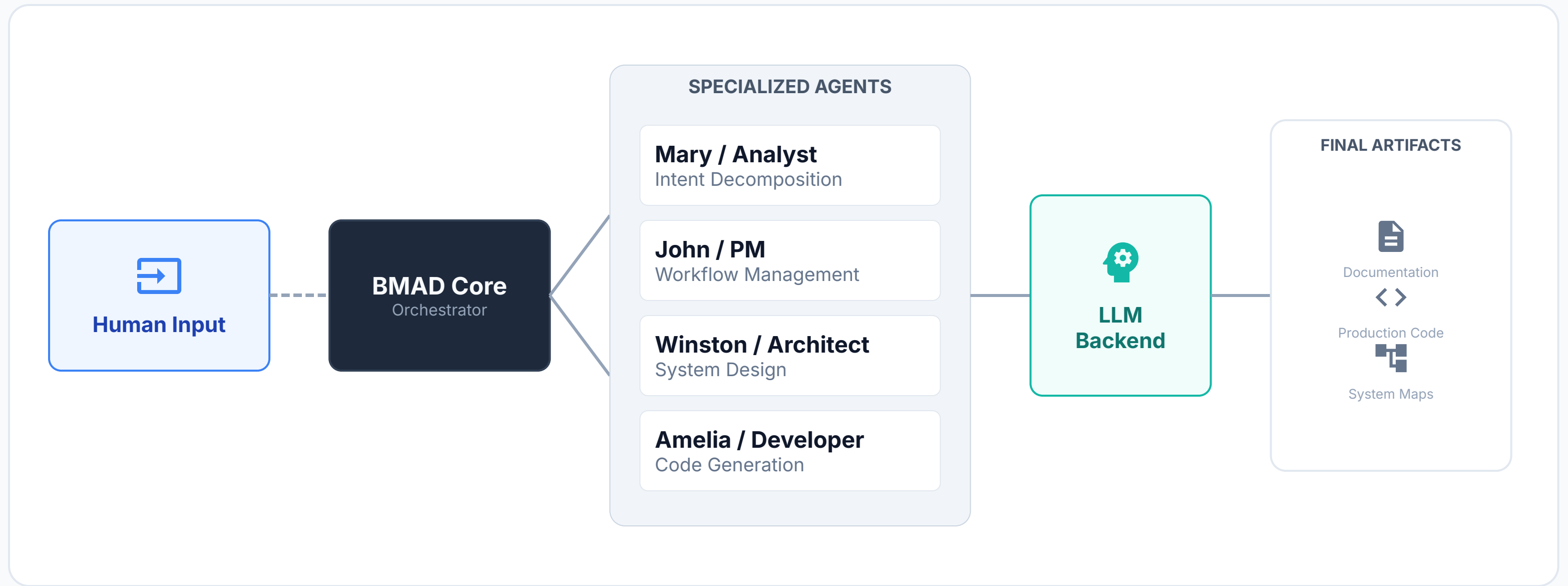


Selected tools:

- GitHub Copilot ☆
- Gemini CLI
- Google Antigravity
- IBM Bob
- Kiro
- OpenCode
- Warp

Agentic Framework Architecture

Methodology over execution: specialized agents in high-quality workflows.



Green Field

Greenfield development via BMAD methodology



Green Field: RPGLE-Json Query Utility

Greenfield development via BMAD methodology

Target & Constraints



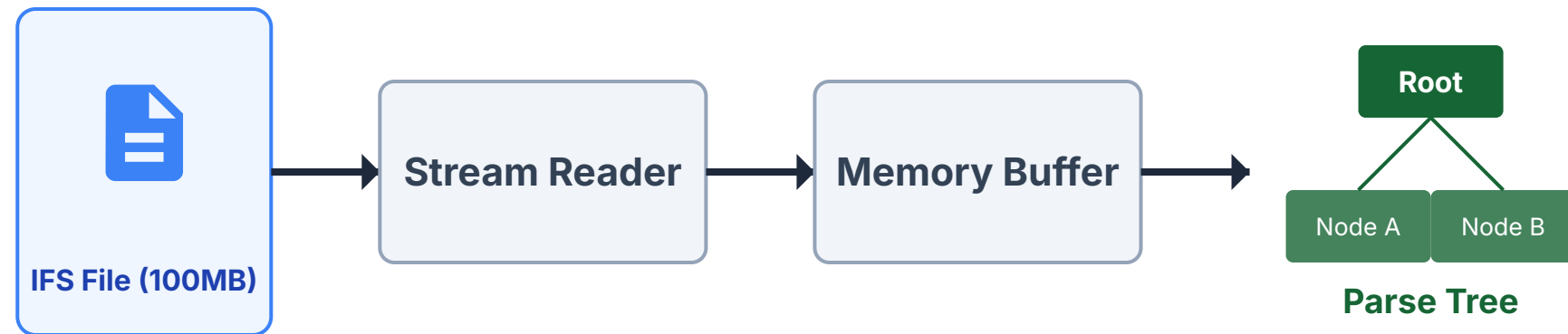
Objectives:

- Fixed-format RPGLE program
- High-performance JSON querying

Constraints:

- Zero external dependencies
- Sub-millisecond response
- Streaming architecture (100MB IFS)

Architectural Data Flow



The architecture utilizes a **Streaming Strategy** to handle large JSON payloads without exhausting system memory.

By decoupling reading from parsing via an intermediate memory buffer, the utility achieves **sub-millisecond latency** for standard queries while maintaining 100% RPGLE compatibility.

Conversation with Mary

Gathering Project Goals & Requirements

Characters

 **Mary:** Business Analyst Agent

 **Mauro:** RPG Developer

Technical Constraints

- Fixed-form RPGLE code
- Zero 3rd party libraries
- OS/400 Version 7.5 Target
- IFS File source (JSON)
- JSON-path query parameter

Mary:

"I'll help you create a product brief for the RPGLE-Json project. To get started, could you tell me what this project is about and what you're trying to accomplish?"

Mauro:

"Let me explain the project concept in my own words."

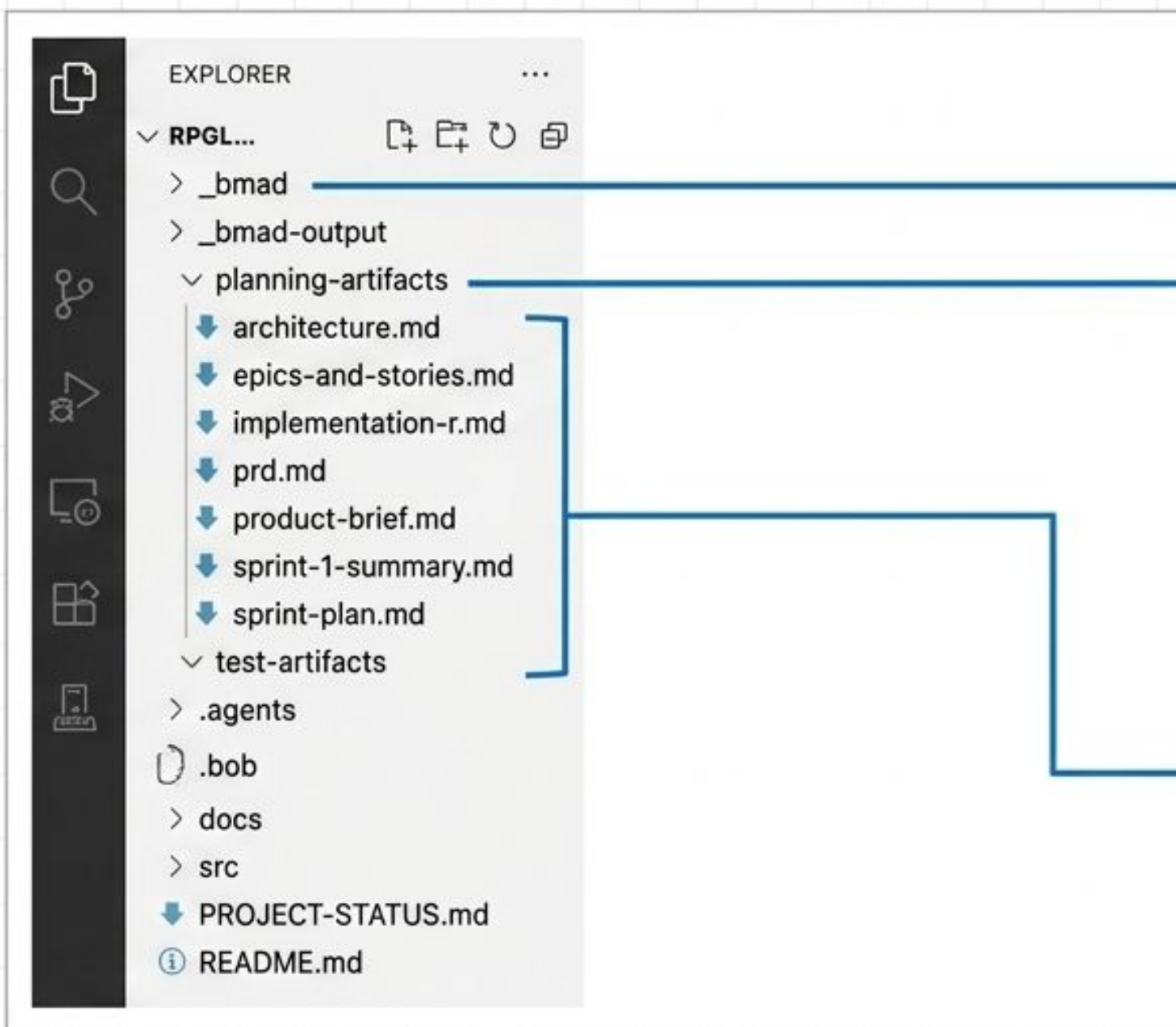
Mary:

"Perfect! Please describe your RPGLE-Json project concept. What problem does it solve, who will use it, and what are the main goals you want to achieve?"

Mauro's Project Definition:

"I want to create a RPG program that reads a RPG file from the IFS and searches inside it using a json-path query as a parameter, then returns the nodes matching the query in form of string variable. The rpgle version should be fixed-form, I don't want to use external 3rd party libraries and it has to work for OS/400 version 7.5."

Establishing the Autonomous Workspace and Artifact Structure



The Operational Core

The `_bmad` directory contains the local framework execution logic.

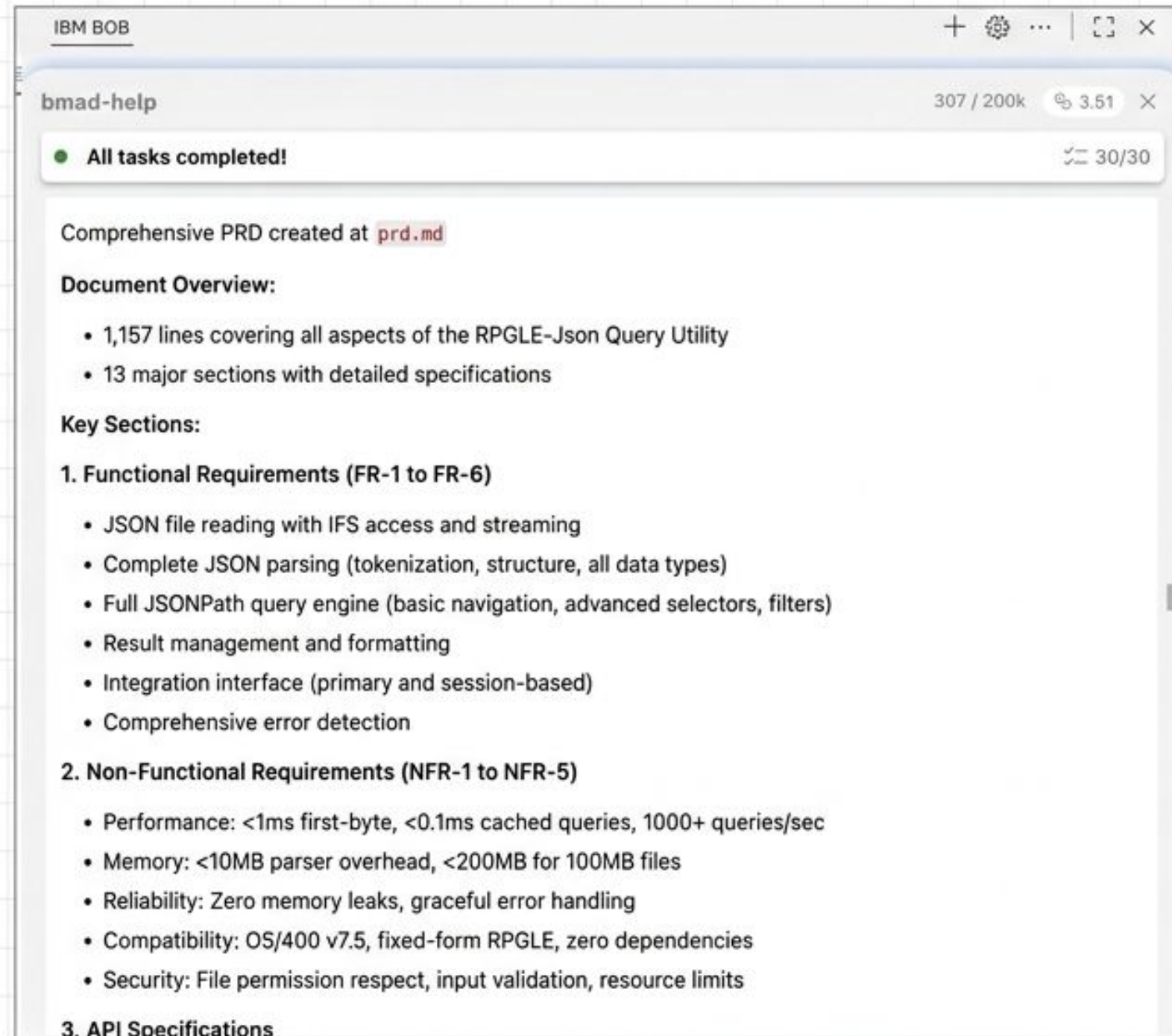
The Repository of Specifications

Generated requirements, architecture, and sprint plans are stored centrally.

Markdown as Source of Truth

Structured markdown files replace transient chat history, ensuring permanent traceability.

Generating the Comprehensive Product Requirements Document



Specification Constraints:

- AI agents independently generate complete Functional and Non-Functional requirements prior to implementation.
- Establishes exact performance targets and memory constraints.
- Creates API specifications and strict acceptance criteria.

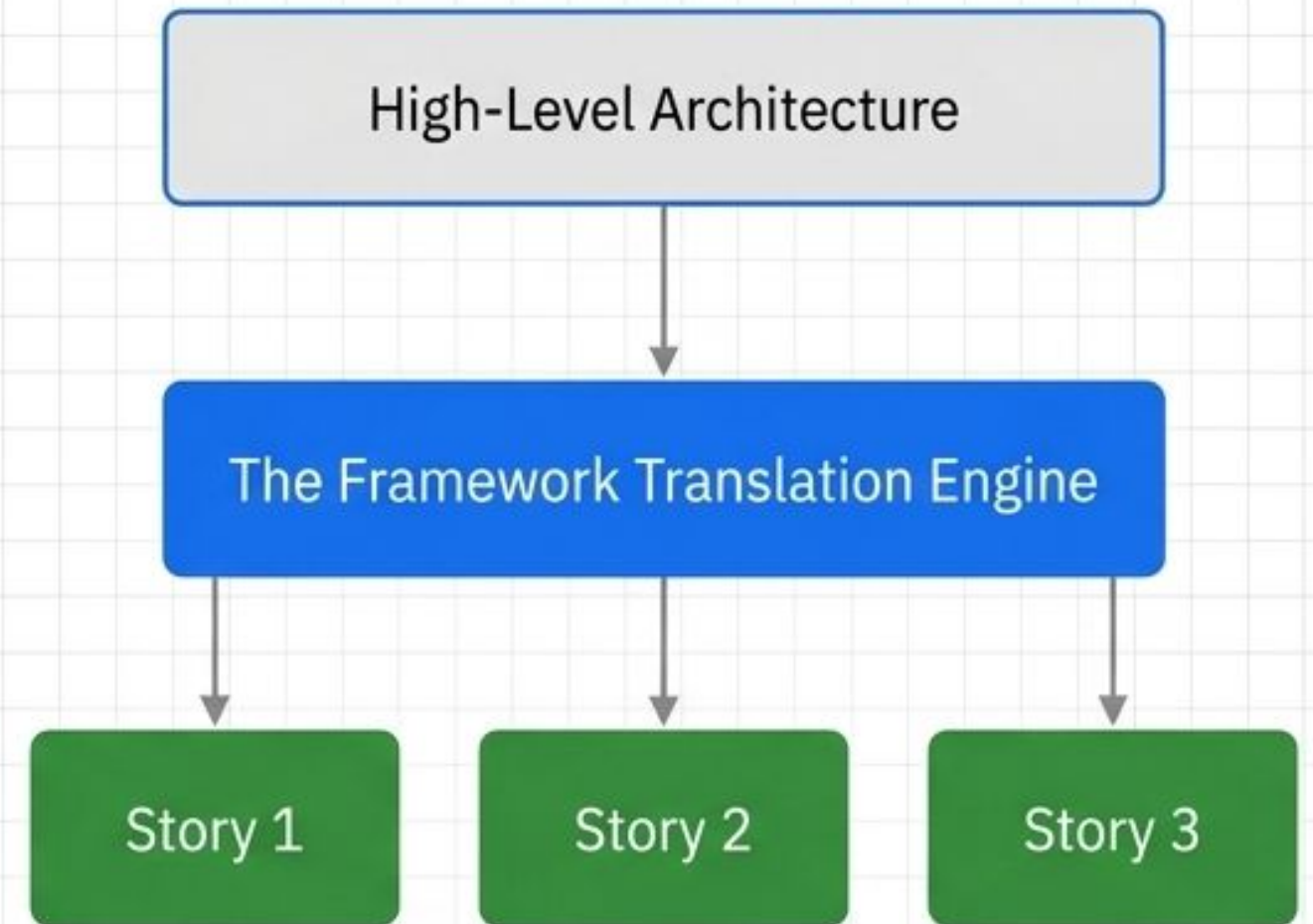
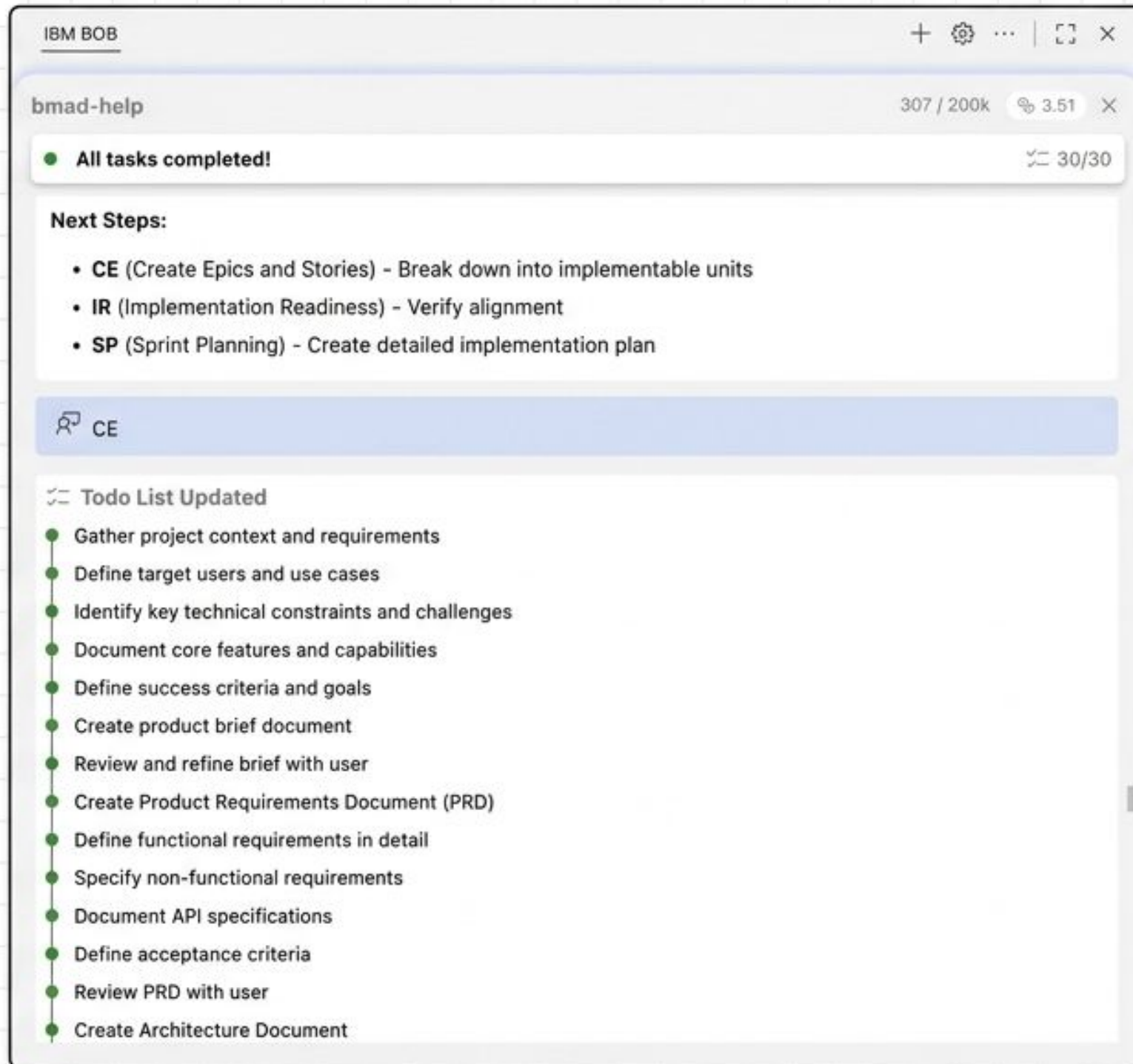
Defining Technical Approaches and System Boundaries

```
product-brief.md prd.md architecture.md x epics-and-stories.mc ...
_bmad-output > planning-artifacts > architecture.md > # RPGLE-Json Query Utility > ## Architecture E
2 # RPGLE-Json Query Utility
13 ## Table of Contents
27
28 ## Architecture Overview
29
30 ### Design Philosophy
31
32 The RPGLE-Json architecture follows these core principles:
33
34 1. **Streaming-First:** Process JSON incrementally to handle large files
efficiently
35 2. **Stateful Sessions:** Cache parsed structures for repeated queries
36 3. **Modular Design:** Separate concerns into distinct, testable
components
37 4. **Fixed-Form Compatible:** Use only fixed-form RPGLE constructs
38 5. **Zero Dependencies:** Implement all functionality natively
39
40 ### High-Level Architecture
41
42 Client RPGLE Program
43 (Uses /COPY JSONORY)
44
45
46
47
48
49
50 Public API Layer
51
52 JSONORY (One-shet)
53 JSONOPEN (Session)
54 JSONCLOSE (Session)
55
```

Non-Negotiable Architecture Rules:

- Architecture documents establish the strict constraints for subsequent developer agents.
- Enforces decisions: Zero Dependencies, Stateful Sessions, and Streaming-First paradigms.
- Maintains strict compatibility with legacy IBM i environments.
- Prevents the model from hallucinating unapproved technical stacks.

Deconstructing Architecture into Executable Sprint Plans



- The framework translates complex architecture into implementable stories.
- Maintains end-to-end traceability from the PRD to the individual code commit.
- Provides the necessary structure for continuous validation and quality assurance loops.

Automated Compliance and Operational Traceability

Continuous Status Reporting

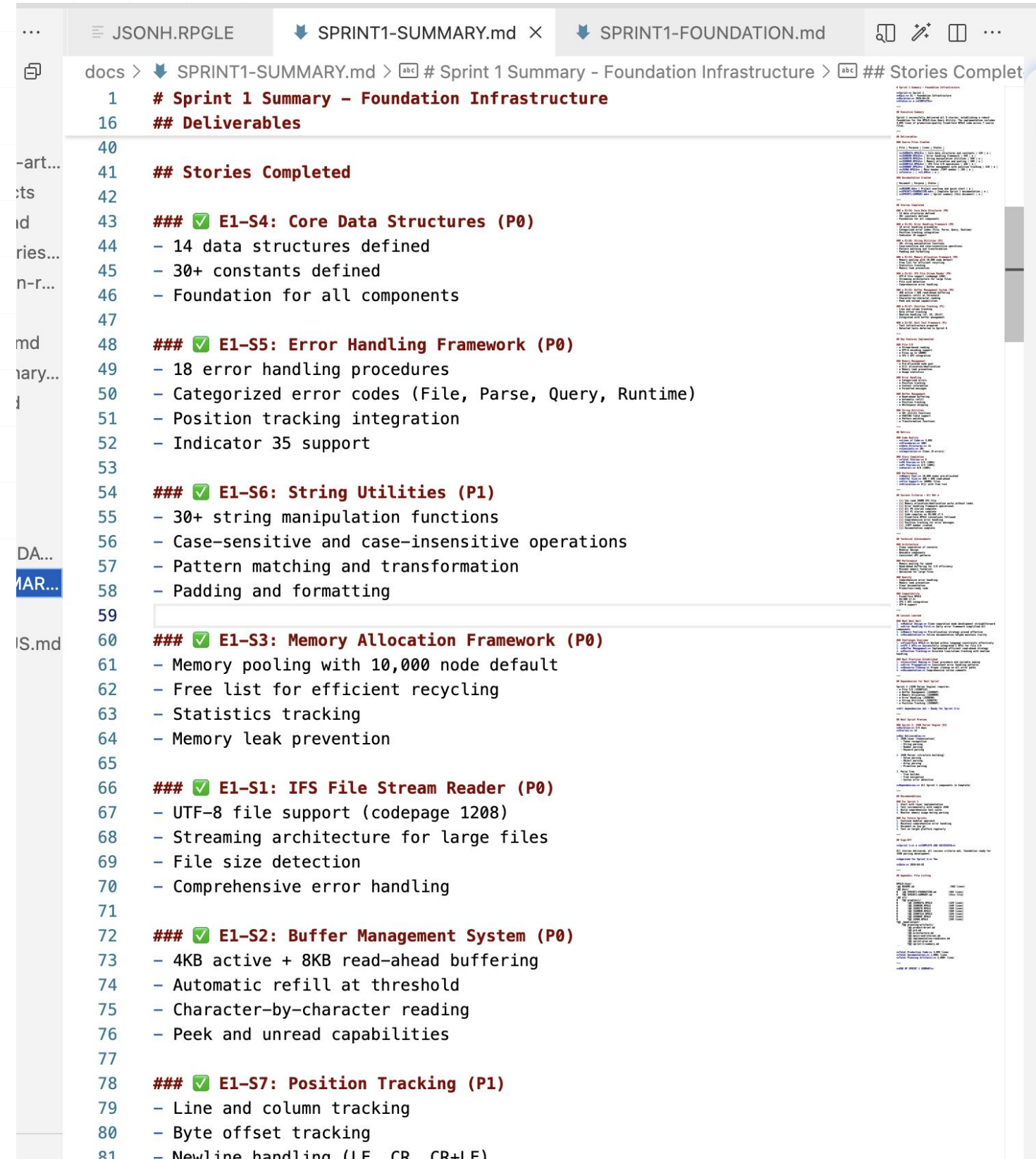
- Status reporting is generated alongside code, not after.
- Satisfies the need for permanent audit trails (NIS, SOC, ISO).
- Updates project status, timelines, and completion metrics autonomously.

```
PROJECT-STATUS.md x JSONH.RPGLE SPRINT1-SUMMARY.md
PROJECT-STATUS.md > # RPGLE-Json Query Utility - Project Status
1 # RPGLE-Json Query Utility - Project Status
303 ## team & resources
312
313 ## Timeline
314
315 | Sprint | Duration | Start | End | Status |
316 |-----|-----|-----|-----|-----|
317 | Sprint 1 | 1 day | 2026-04-28 | 2026-04-28 | ✔ Complete |
318 | Sprint 2 | 3-5 days | TBD | TBD | 🚧 Planned |
319 | Sprint 3 | 3-5 days | TBD | TBD | 🚧 Planned |
320 | Sprint 4 | 2-3 days | TBD | TBD | 🚧 Planned |
321 | Sprint 5 | 2-3 days | TBD | TBD | 🚧 Planned |
322 | Sprint 6 | 2-3 days | TBD | TBD | 🚧 Planned |
323 | **Total** | **14-22 days** | | | **12.3% Complete** |
324
325 ---
326
327 ## Contact & Links
328
329 **Repository:** RPGLE-Json
330 **Documentation:** [README.md] (README.md)
331 **Sprint 1 Details:** [docs/SPRINT1-FOUNDATION.md] (docs/SPRINT1-FOUNDATION.md)
332 **Sprint 1 Summary:** [docs/SPRINT1-SUMMARY.md] (docs/SPRINT1-SUMMARY.md)
333
334 ---
335
336 ## Change Log
337
338 ### 2026-04-28
339 - ✔ Sprint 1 complete
340 - ✔ All foundation components implemented
341 - ✔ Documentation complete
342 - ✔ Ready for Sprint 2
343
344 ---
345
346 **Project Status:** ✔ **ON TRACK**
347 **Current Phase:** Foundation Complete, Parser Development Next
348 **Overall Progress:** 12.3% (8/65 stories)
349
350 ---
351
352 **END OF PROJECT STATUS**
```

Automated Compliance and Operational Traceability

Continuous Status Reporting

- Status reporting is generated alongside code, not after.
- Satisfies the need for permanent audit trails (NIS, SOC, ISO).
- Updates project status, timelines, and completion metrics autonomously.



The screenshot shows a code editor with a file named 'SPRINT1-SUMMARY.md'. The content is a markdown document titled '# Sprint 1 Summary - Foundation Infrastructure' with a sub-header '## Deliverables'. It lists several completed items, each with a green checkmark icon:

- ### E1-S4: Core Data Structures (P0)
 - 14 data structures defined
 - 30+ constants defined
 - Foundation for all components
- ### E1-S5: Error Handling Framework (P0)
 - 18 error handling procedures
 - Categorized error codes (File, Parse, Query, Runtime)
 - Position tracking integration
 - Indicator 35 support
- ### E1-S6: String Utilities (P1)
 - 30+ string manipulation functions
 - Case-sensitive and case-insensitive operations
 - Pattern matching and transformation
 - Padding and formatting
- ### E1-S3: Memory Allocation Framework (P0)
 - Memory pooling with 10,000 node default
 - Free list for efficient recycling
 - Statistics tracking
 - Memory leak prevention
- ### E1-S1: IFS File Stream Reader (P0)
 - UTF-8 file support (codepage 1208)
 - Streaming architecture for large files
 - File size detection
 - Comprehensive error handling
- ### E1-S2: Buffer Management System (P0)
 - 4KB active + 8KB read-ahead buffering
 - Automatic refill at threshold
 - Character-by-character reading
 - Peek and unread capabilities
- ### E1-S7: Position Tracking (P1)
 - Line and column tracking
 - Byte offset tracking
 - Newline handling (LF, CR, CR+LF)

Automated Compliance and Operational Traceability

Continuous Status Reporting

- Status reporting is generated alongside code, not after.
- Satisfies the need for permanent audit trails (NIS, SOC, ISO).
- Updates project status, timelines, and completion metrics autonomously.

```
488 ---
489
490 ### Day 3: Polish & Testing
491 **Focus:** Complete P1 stories and comprehensive testing
492
493 **Morning (4 hours):**
494 1. **E1-S7: Position Tracking** (2 hours)
495 |   - Implement position tracking
496 |   - Test with various inputs
497 |   - Integrate with buffer management
498
499 2. **E1-S8: Unit Test Framework** (2 hours)
500 |   - Create test harness
501 |   - Implement assertions
502 |   - Write initial tests
503
504 **Afternoon (4 hours):**
505 3. **Integration Testing** (2 hours)
506 |   - Test file I/O with large files
507 |   - Test memory allocation under load
508 |   - Test error handling scenarios
509
510 4. **Documentation & Cleanup** (2 hours)
511 |   - Document all procedures
512 |   - Clean up code
513 |   - Prepare for Sprint Review
514
515 **End of Day Goals:**
516 | - [ ] All 8 stories complete
517 | - [ ] Comprehensive test coverage
518 | - [ ] Documentation complete
519 | - [ ] Ready for Sprint 2
520
521 **Validation:**
```

Sprint 1 Issue Identification: Format Compliance

Architecture Overview

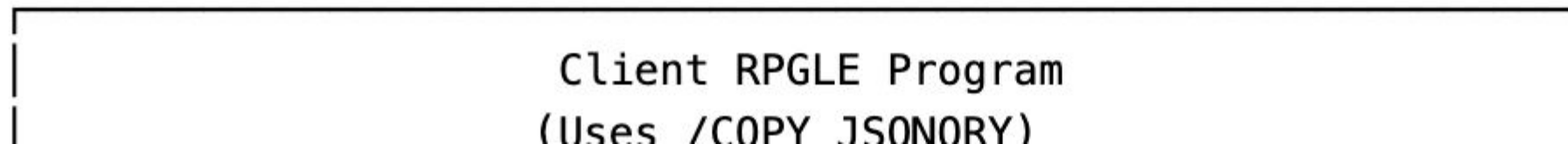
Design Philosophy

The RPGLE-Json architecture follows these core principles:

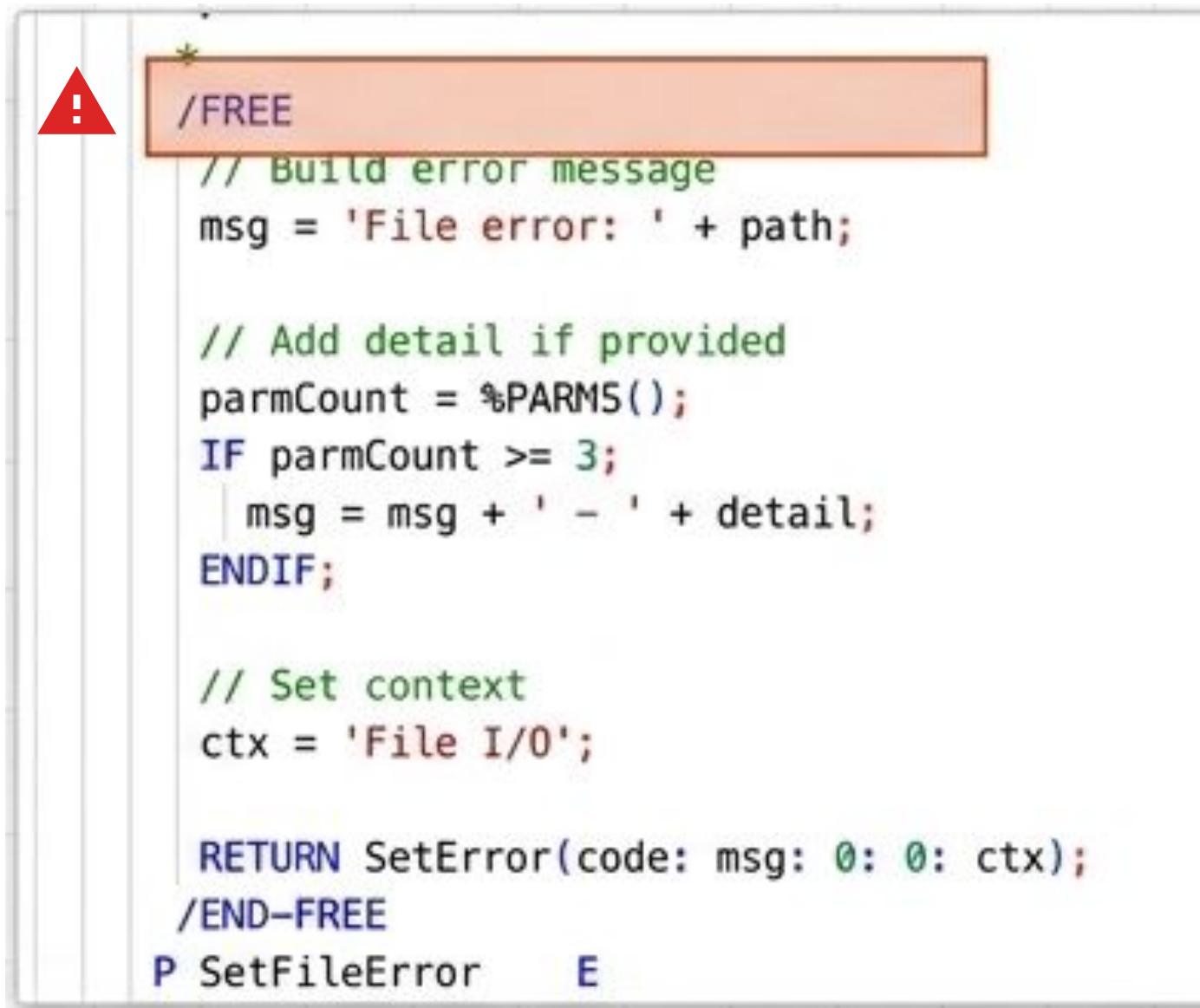
1. ****Streaming-First:**** Process JSON incrementally to handle large files efficiently
2. ****Stateful Sessions:**** Cache parsed structures for repeated queries
3. ****Modular Design:**** Separate concerns into distinct, testable components
4. ****Fixed-Form Compatible:**** Use only fixed-form RPGLE constructs
5. ****Zero Dependencies:**** Implement all functionality natively

High-Level Architecture

...



Sprint 1 Issue Identification: Format Compliance



```
*  
/FREE  
// Build error message  
msg = 'File error: ' + path;  
  
// Add detail if provided  
parmCount = %PARMS();  
IF parmCount >= 3;  
| msg = msg + ' - ' + detail;  
ENDIF;  
  
// Set context  
ctx = 'File I/O';  
  
RETURN SetError(code: msg: 0: 0: ctx);  
/END-FREE  
P SetFileError      E
```

The Violation

The implementation agent utilized the modern `/FREE` format syntax, **breaching the fixed-format rule** established for this specific legacy enterprise environment.

The Root Cause

The underlying language model mathematically evaluates multiple valid architectural pathways. Without explicit, heavily documented constraints, the **LLM defaults to its broader training distribution** (modern syntax) rather than the highly specific legacy compatibility structures required.

Fixed with Antigravity CLI



Antigravity CLI 1.0.7

mauro.sanfilippo@smeup.com (Antigravity Starter Quota)

Gemini 3.5 Flash (Medium)

~/works/RPGLE-Json

```
> using BMAD, help me fix a misunderstanding in the last sprint. I instructed to use ONLY Fixed Form that allow to use the free-form RPG. This is not allowed. The problem is "we cannot use free-form"
```

```
I will start by listing the files in the directory to see the project's layout and find any BMAD
```

```
● ListDir(/Users/mauro/works/RPGLE-Json) (ctrl+o to expand)
```

```
I will view the PROJECT-STATUS.md file to understand the current sprint status, epics, and who
```

```
● Read(/Users/mauro/works/RPGLE-Json/PROJECT-STATUS.md) (ctrl+o to expand)
```

```
I will search for the /FREE directive in the src/ directory to locate where free-form RPG b
```

```
● Search(/FREE in /Users/mauro/works/RPGLE-Json/src) (ctrl+o to expand)
```

```
▶ Thought for 1s, 546 tokens
```

```
Examining Code Structure
```

Why Antigravity + Gemini?

The Antigravity CLI provides the necessary agentic reasoning to bridge legacy constraints with modern AI capabilities.

- Autonomous plan correction
- Constraint-aware code generation
- Real-time auditability

BMAD is independent!

The framework operates as a standalone orchestrator, ensuring that model-specific biases (like defaulting to /FREE format) are corrected before deployment.

Enforcing Architectural Course Correction

Decision 5: STRICT Fixed-Form RPGLE Mandate

****Decision:**** ALL RPGLE source code and architectural designs MUST use strict fixed-format (H, F, D, I, C, O, P specs). The use of `/FREE` or `/END-FREE` blocks is STRICTLY PROHIBITED.

****Rationale:****

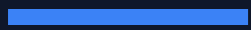
- ✓ Ensures 100% compatibility with legacy systems and established coding standards.
- ✓ Prevents fragmentation of coding styles within the project.
- ✓ Forces disciplined use of traditional RPGLE constructs.

Resolution: Deterministic Constraint Modification

- *"Good design and algorithmic ideas are frequently crafted **while coding**."*
- The [architecture.md](#) document is updated with a **STRICT fixed-form mandate**.
- The framework reads this update and forces all subsequent developer agents to comply **deterministically**.

Brown Field

Brownfield development via BMAD methodology



Brown Field: Sample ERP

Brown field development via BMAD methodology

```
SAMMNU                               Arcad Sample Application
                                     System:      VP26X026

Select one of the following:

Master files
  1. Work with Articles                ART200
  2. Work with Customers              CUS200
  3. Work with Customer Orders       ORD201
  4. Work with Providers              PR0200
  5. Display Providers               PR0201
  6. Create a Customer Order.        ORD100

Reports
  10. Article to purchase              PR0203
  11. Customer with Open Order       QMQRV:CUSQRV
  12. Article by Last Order Date     QMQRV:ARTQRV

                                     More...

Selection or command
===> _____

F3=Exit   F4=Prompt   F6=Messages   F9=Retrieve   F10=Job log   F12=Cancel
F14=Submitted jobs

MA  A                                     20/007
```

Brown Field: Sample ERP

Brown field development via BMAD methodology

```
ART200-1                               Work with Articles                               6/12/26
Position to . . .                       00:28:25
Type options, press Enter.
      2=Edit      3=Info      4=Delete      6=Suppliers

Opt  Id      Description                               Fam  Del
___  ___
___  000004  Bluetooth Headphones                               ELE
___  000042  Board Game Family Edition                         TOY
___  000012  Bookshelf 5 Shelves                               FUR
___  000040  Building Blocks Set 500pcs                       TOY
___  000031  Business Management Handbook                     BOO
___  000072  Chocolate Gift Box Assorted                       FOO
___  000081  Coffee Maker Automatic                           HOM
___  000013  Coffee Table Modern                               FUR
___  000011  Desk 120x60cm White                              FUR
___  000051  Dumbbell Set 20kg                                SPO
___  000032  Fiction Novel Bestseller                          BOO
___  000060  Garden Hose 25m                                  GAR
___  000062  Garden Tool Set 5pcs                             GAR
___  000022  Jacket Winter XL                                 CLO

More...
```

F3=Exit F6=Create F12=Cancel

MA A 02/023

Brown Field: Sample ERP

Brown field development via BMAD methodology

```
CUS200-1                               Work with Customers                               6/12/26
Position to name . . .                   00:28:42
Type options, press Enter.
    2=Edit      5=Orders

Opt Id      Customer                      City                      Del
-----
   1 Acme Corporation                      Paris
   5 Distribution Plus                    Toulouse
   6 Enterprise Systems                   Nantes
   2 Global Trading SA                    Lyon
   8 Premium Buyers Group                  Lille
   7 Quality Goods SARL                   Strasbourg
   4 Retail Masters Inc                    Bordeaux
   9 Smart Commerce                       Rennes
   3 Tech Solutions Ltd                    Marseille
  10 Value Partners SA                     Dijon

Bottom

F3=Exit      F5=Refresh      F6=Create      F12=Cancel

Mâ  A  02/027
```

Brown Field: Sample ERP

Brown field development via BMAD methodology

```
CUS301                               Select a Customer
Name contains . . . : _____
City contains . . . : _____
Make a selection. Press Enter
1=Select

Opt Code Name                               City                               FR
 1      1 Acme Corporation                    Paris                               FR
-      5 Distribution Plus                    Toulouse                            FR
-      6 Enterprise Systems                  Nantes                              FR
-      2 Global Trading SA                   Lyon                                FR
-      8 Premium Buyers Group                 Lille                               FR
-      7 Quality Goods SARL                  Strasbourg                          FR
-      4 Retail Masters Inc                   Bordeaux                             FR
-      9 Smart Commerce                       Rennes                              FR
-      3 Tech Solutions Ltd                   Marseille                           FR
-     10 Value Partners SA                    Dijon                               FR

Bottom

F3=Exit      F12=Cancel

M^  A                                           09/003
```

Brown Field: Sample ERP

Brown field development via BMAD methodology

Without Proper Specs

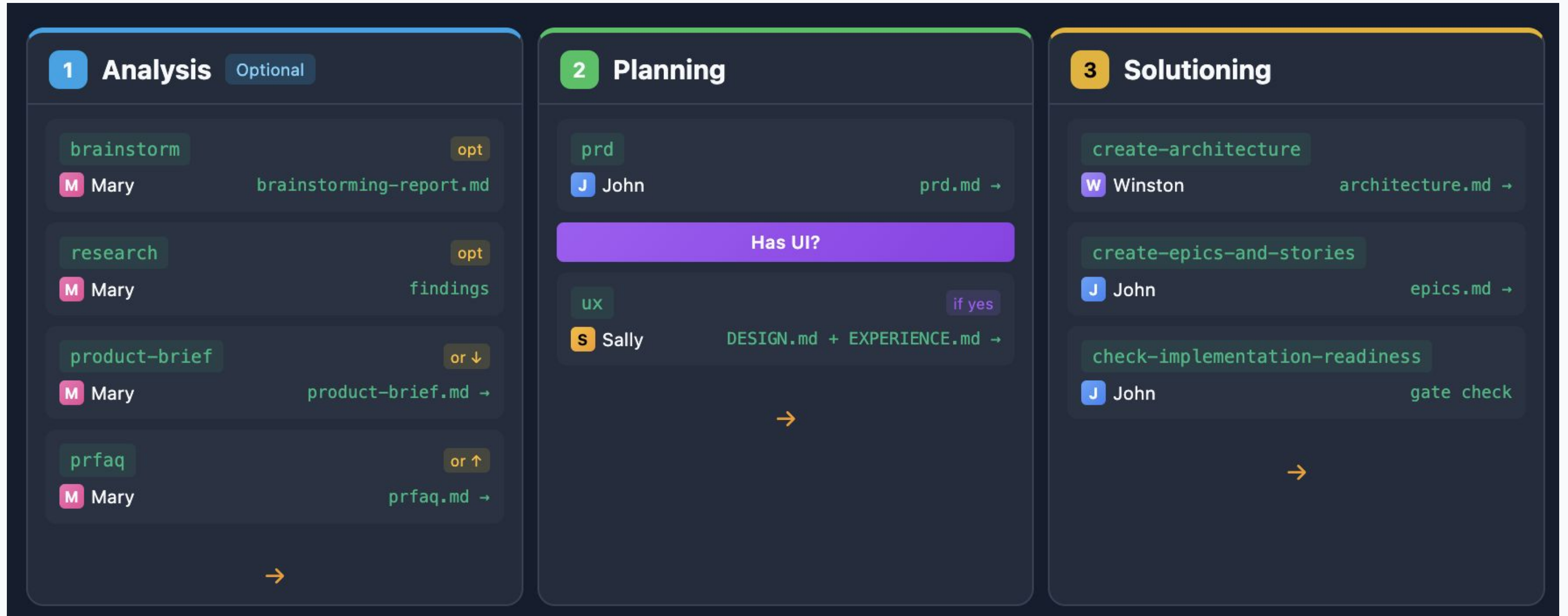
- **Assumptions & guesswork** — AI fills in missing requirements.
- **Context drift** — Focus shifts away from the original task.
- **Unnecessary file edits** — Related but irrelevant code gets touched.
- **Higher review burden** — More time spent validating output.

With Proper Specs (BMAD)

- **Clear objectives** — AI executes against defined requirements.
- **Task alignment** — Changes stay within scope.
- **Targeted changes** — Only required files are modified.
- **Faster reviews** — Git diffs are smaller and easier to verify.

Brown Field: Sample ERP

Brown field development via BMAD methodology



Brown Field: Sample ERP

Brown field development via BMAD methodology

Quick Start

Since you have multiple stories awaiting review, I recommend:

Run Code Review on story 1-2 (oldest in review):

```
1 bmad-code-review
```

Would you like me to start the code review process for story 1-2 now?

Other Available Tools

Anytime Skills:

- [SS] **Sprint Status** - Current sprint summary
- [CK] **Checkpoint** - Review commits/branches/PRs
- [QA] **QA Automation** - Generate E2E tests
- [CC] **Correct Course** - Navigate significant changes
- [WD] **Write Document** - Tech writing assistance

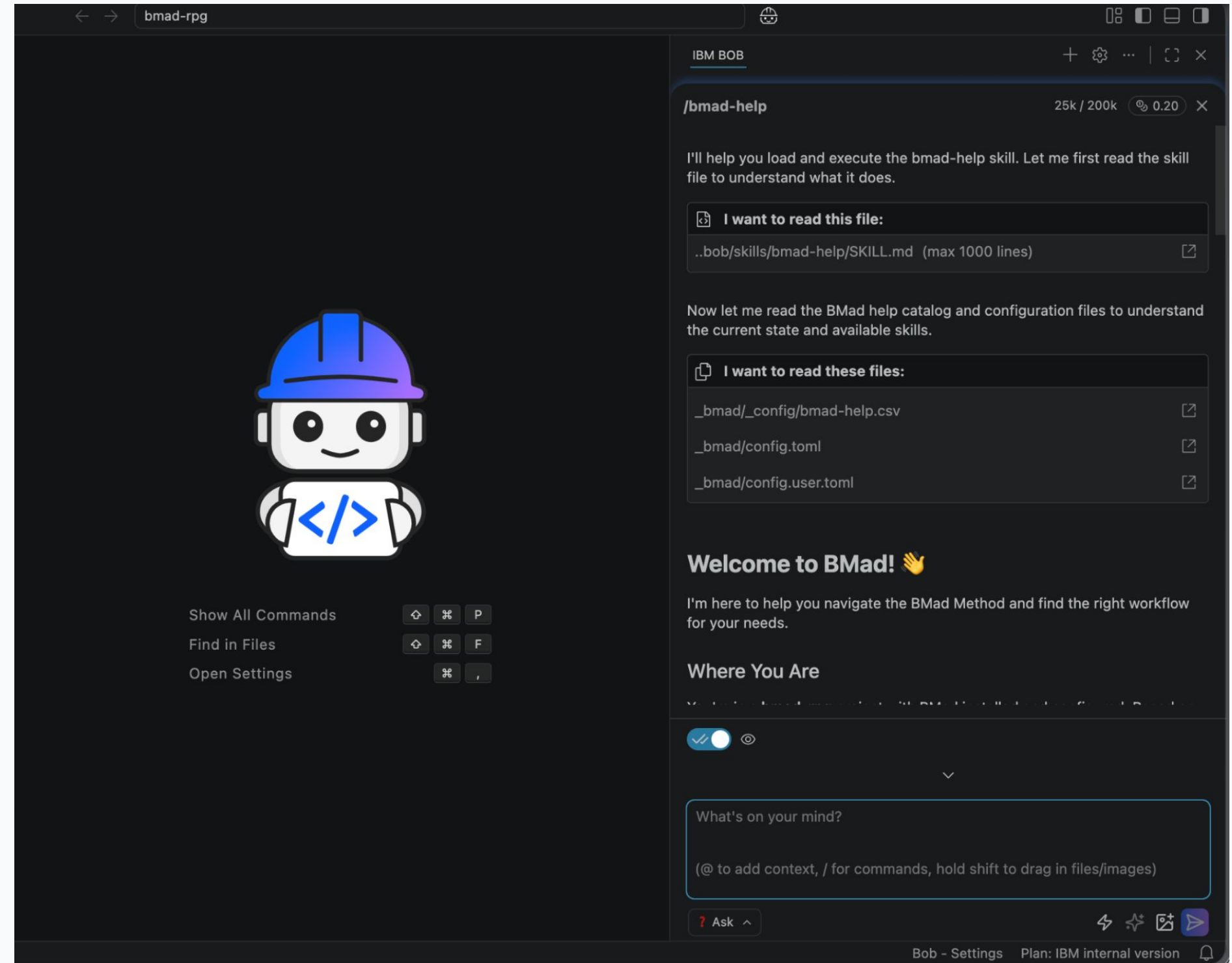
For questions about BMad itself: Ask me directly - I can search BMad document

You can ask follow up questions or use the /clear command to start a new task

> Enter your prompt, / for commands, @ for files, ! for Shell mode

Auto-approve: Off

Tokens



IBM BOB

/bmad-help 25k / 200k 0.20

I'll help you load and execute the bmad-help skill. Let me first read the skill file to understand what it does.

I want to read this file:

- ..bob/skills/bmad-help/SKILL.md (max 1000 lines)

Now let me read the BMad help catalog and configuration files to understand the current state and available skills.

I want to read these files:

- _bmad/_config/bmad-help.csv
- _bmad/config.toml
- _bmad/config.user.toml

Welcome to BMad! 🙌

I'm here to help you navigate the BMad Method and find the right workflow for your needs.

Where You Are

What's on your mind?

(@ to add context, / for commands, hold shift to drag in files/images)

? Ask

Bob - Settings Plan: IBM internal version

Brown Field: Sample ERP

Brown field development via BMAD methodology

Project Context

- **Better context** — Architecture and rules are documented.
- **Higher accuracy** — Fewer assumptions, better code changes.
- **Less drift** — Keeps AI focused on scope.
- **Limited context window** — AI cannot see the entire codebase at once.

- Creates: PRFAQ document in `_bmad-output/planning-artifacts/`

Understanding an Existing Project

[GPC] Generate Project Context — `bmad-generate-project-context`

- Scan your SAMCO codebase to create LLM-optimized documentation
- Essential for brownfield projects like yours
- Creates: project context in `_bmad-output/`

[DP] Document Project — `bmad-document-project`

- Analyze existing project to produce useful documentation
- Creates: documentation in `docs/`

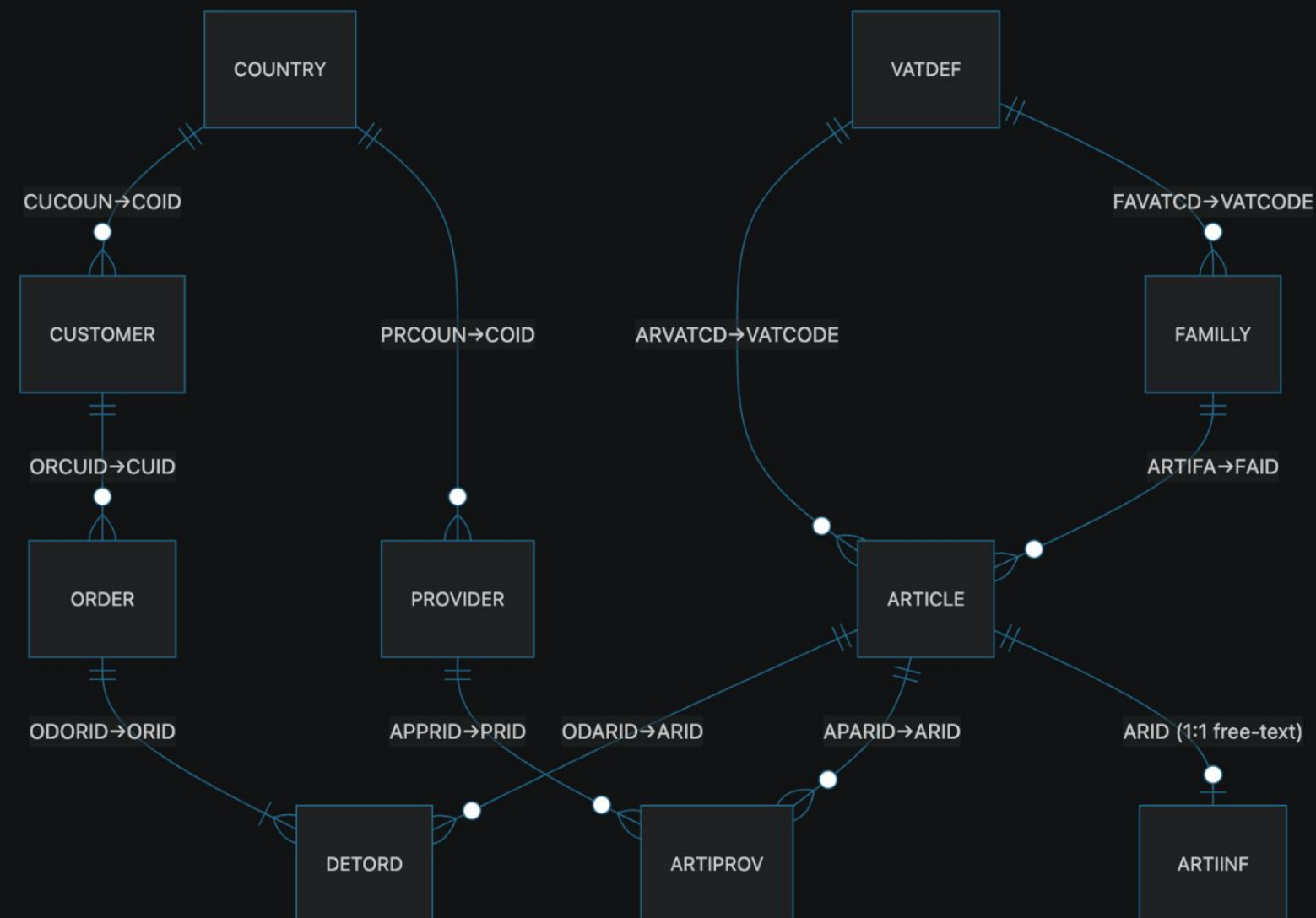
Brown Field: Sample ERP

Brown field development via BMAD methodology

SAMCO — Data Models (Database Schema)

Source of truth: DDS members in SAMCO/QDDSSRC/, SAMCO/common/SAMREF.PF, SAMCO/functionsVAT/vatdef.pf, and SQL objects in SAMCO/QSQLSRC/. Field attributes are inherited from the field-reference file SAMREF unless noted. There are **no DDS-declared referential constraints** — all relationships are application-enforced.

1. Entity-Relationship Overview

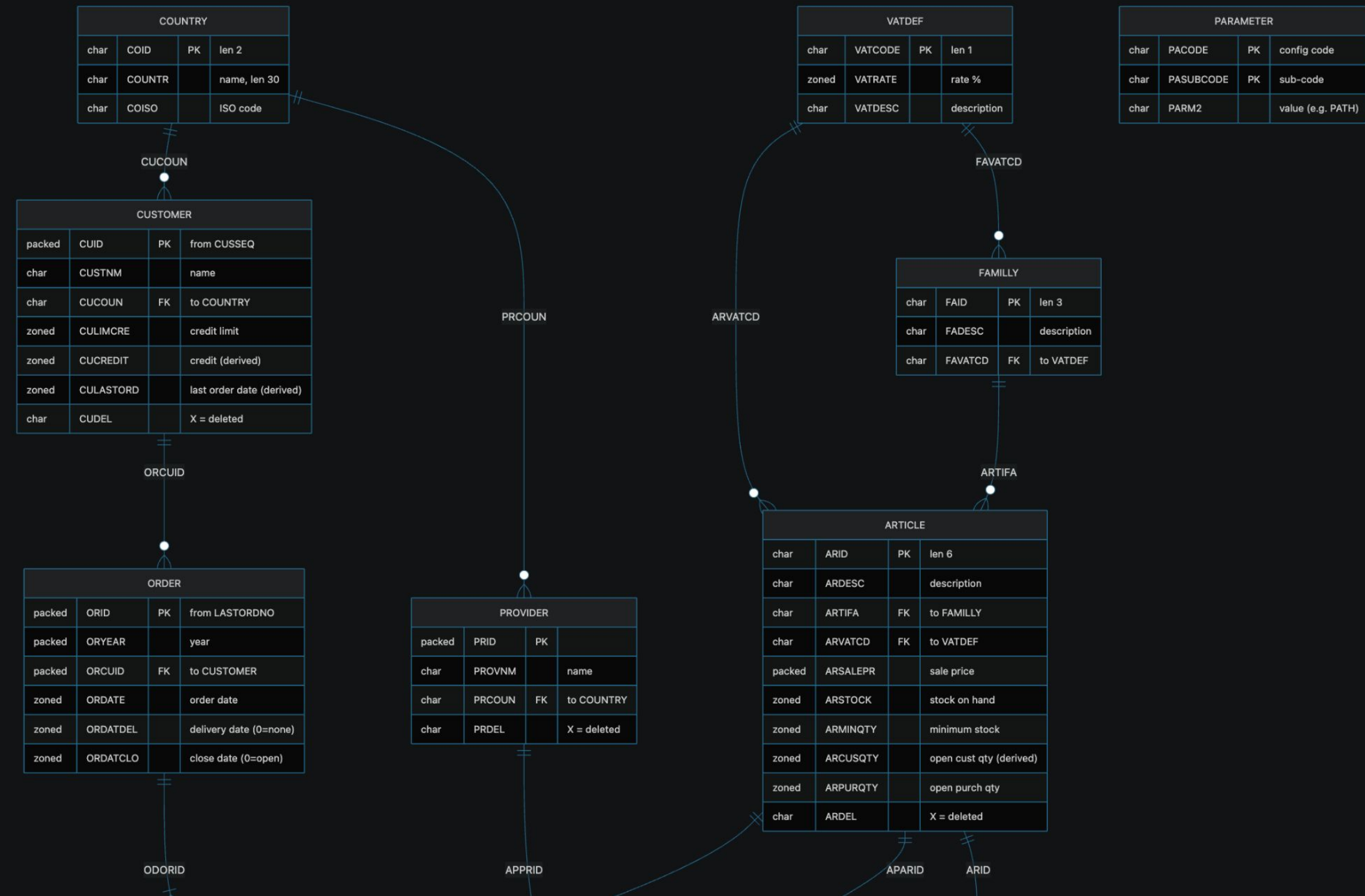


PARAMETER is a standalone key/value configuration table with no relationships.

Brown Field: Sample ERP

Brown field development via BMAD methodology

Attribute-level ER diagram (keys + foreign keys)



Brown Field: Sample ERP

Brown field development via BMAD methodology

SAMCO — Development & Build Guide

Prerequisites

- IBM i partition with Db2 for i, ILE compilers (RPG, COBOL, C/C++, SQL precompilers).
- Tobi / `makei` (the "bob" Better Object Builder) installed at `/QOpenSys/pkgs/bin/makei`.
- Source pushed to the IFS on the IBM i (the toolchain compiles from IFS stream-file source members).
- A target library — referenced as `&lib1` (`objlib/curlib`) in `iproj.json`.

Project configuration

`SAMCO/iproj.json` (Tobi project descriptor):

```
{
  "objlib": "&lib1",
  "curlib": "&lib1",
  "includePath": ["includes", "QPROTOSRC"],
  "preUsrLib": ["&lib1"],
  "compileCommand": "/QOpenSys/pkgs/bin/makei c -f {filename}",
  "buildCommand": "/QOpenSys/pkgs/bin/makei build"
}
```

- `includePath` puts `includes/` and `QPROTOSRC/` on the `/COPY` resolution path (so prototype members resolve without library qualification).
- Per-directory `.ibmi.json` sets the source `tgtCcsid`: `root 37`, `RPGLE/SQL/SRV/PROTO/TRG 297`, `CL/COBOL/VAT 500`. Match these when editing/creating members or the compiler will mis-read accented characters.

Build system

Build is declarative via `Rules.mk` files. The top-level `SAMCO/Rules.mk` lists `SUBDIRS`; each subdirectory's `Rules.mk` declares `TARGET.TYPE`: `prerequisites` rules (e.g. `VAT300.MODULE: vat300.rpgle QPROTOSRC/vat.rpgleinc VATDEF.FILE`). `makei` derives the dependency graph and runs the right IBM i create commands.

Commands:

```
makei build           # build the whole application (dependency-ordered)
makei c -f <filename> # compile a single source member
```

Brown Field: Sample ERP

Brown field development via BMAD methodology

SAMCO — Component Inventory

A catalog of every runnable/UI component, grouped by layer. Naming: interactive programs `<ENTITY><nnn>` (e.g. ORD200), service modules `<ENTITY>3xx` (e.g. ART300), service programs `F<ENTITY>` (e.g. FARTICLE), display files `<pgm>D`, printer files `<pgm>0`, CL wrappers `<pgm>C`.

1. Interactive (5250) programs — `QRPGLESRC/`, `QRPGSRC/`

All ILE programs share a uniform **subfile state machine**: a `panel` variable selects a format subroutine, and per-panel `step` variables run a `prp → lod → dsp → key → chk → act` cycle. Function keys read from the display `INDDS`.

Program	Type	Screen	Purpose
ART200	SQLRPGLE	ART200D	Work with Articles — list/create/change/delete, edit free-text info (ARTIINF), jump to providers (→ART201). Auto-generates ARID.
ART201	RPGLE	ART201D	Article→providers links (article-side of ARTIPROV).
ART202	RPGLE	ART202D	Provider→articles links (provider-side of ARTIPROV).
CUS200	SQLRPGLE	CUS200D	Work with Customers — new ID from CUSSEQ; validates country; jump to orders (→ORD200).
ORD100	RPGLE	ORD100D	Create order — builds lines in QTEMP/DETORD, computes line totals + VAT, on F8 confirm assigns ORID from LASTORDNO, writes header+lines, prints (→ORD500).
ORD101	RPGLE	ORD101D	Edit order lines / record deliveries; qty-vs-delivered validations.
ORD200	SQLRPGLE	ORD200D	Orders for one customer (cursor over ORDERCUS). Options: edit/delete/display/print/close/deliver.
ORD201	SQLRPGLE	ORD201D	Work with all orders (paged). Menu option 3.
ORD202	RPGLE	ORD202D	Display one order (read-only) with lines + totals.
ORD500	RPGLE	ORD5000 (PRTF)	Print order document; calls ORD500C for PDF. (<i>Not interactive.</i>)
ORD700	RPGLE	—	DETORD trigger program — maintains ARTICLE.ARCUSQTY on line insert/update/delete; logs deletes.
ORD900	RPGLE	—	Utility — reseed LASTORDNO from max order (menu 80).
ORD901	SQLRPGLE	—	Batch — shift all order dates toward today; refresh DETORD.ODYEAR and CUSTOMER.CULASTORD (menu 81).
PAR200	RPGLE	PAR200D	Work with Parameters (PARAMETER table CRUD).

Brown Field: Sample ERP

Brown field development via BMAD methodology

SAMCO — Source Tree Analysis

The codebase is an IBM i source tree organized by **source type** (the classic Q*SRC library-member convention), built with **Tobi/makei**. The application lives under **SAMCO/**; the repository root also carries the BMad tooling (**_bmad/**, **_bmad-output/**, **docs/**).

```
bmad-rpg/          # repo root (BMad workspace)
├── SAMCO/         # ← the IBM i application
│   ├── iproj.json # Tobi project (objlib &lib1, includePath includes/ + QPROTOSRC/)
│   ├── .ibmi.json # build CCSID (root = 37)
│   ├── Rules.mk   # top-level build: SUBDIRS list
│   ├── README.md  # project intro (modernization-with-Bob demo)
│   ├── SAMCO_Architecture_Documentation.md # prior hand-written arch doc (dated)
│   └── POPULATE_SAMCO_TABLES.sql # full-library seed data
│
│   ├── common/    # SAMREF.PF – field reference file (built first)
│   ├── functionsVAT/ # self-contained VAT service: vatdef.pf, vat300.rpgle, vat.rpgleinc, fvat.bnd
│   ├── globalization/ # NLS demos: CHS/ DEU/ HEB/ (CCSID/bidi source tests)
│   └── includes/   # included.clle (shared CL DCL fragment)
│
│   ├── QDDSSRC/   # * DDS: 9 PF, 12 LF, 18 DSPF, 1 PRTF (data + UI definitions)
│   ├── QRPGLSRC/  # * ILE RPG: interactive pgms, *3xx service modules, utilities
│   ├── QRPGSRC/   # legacy fixed-form RPG (COU200 only)
│   ├── QPROTOSRC/ # * /COPY prototypes (function contracts) – on includePath
│   ├── QSRVSRV/   # binder source (.BND export lists) for service programs
│   ├── QILESRSRC/ # service-program build specs (.ILESrvPGM → CRTSRVPGM)
│   ├── QILESRC/   # ILE program build specs (.ILEPGM)
│   ├── QBNSRC/    # SAMPLE.BNDDIR (binding directory, 12 entries)
│   ├── QSQRCSRC/  # * SQL: views, proc (ART801), UDFs, sequence, trigger, table, seed
│   ├── QTRGSRC/   # system PF trigger registrations (ORD700A/D/U on DETORD)
│   ├── QDTASRC/   # data area LASTORDNO, data queue STREAMDTA
│   ├── QCLSRC/    # CL programs (ORD100C/C2, ORD500C, PAR201, OPM)
│   ├── QCMDSRC/   # commands (CRTORD, CVTSPLPDF)
│   ├── QMSGSRC/   # message files (SAMMSGF, SGMSGF)
│   ├── QPNLSRC/   # SAMMNU menu, SAMHELP help, wstrig workstation cust.
│   ├── QCBLSRC/   # ILE COBOL (PR0201 work-with-providers; verify* samples)
│   ├── QCSRC/ QCPPSRC/ # ILE C/C++ (PUTIFS, PWD)
│   └── QSQLCSRC/ QSQLCPPSRC/ # SQL C/C++ (ANZ_FILE analyze-file tool)
│
│   ├── .evfevent/ .logs/ # build event/log artifacts (gitignored)
│   └── .env .ibmi.json # local env / build config
│
├── _bmad/         # BMad method tooling & module configs
├── _bmad-output/ # planning & implementation artifacts (BMad)
├── docs/         # ← generated documentation (this folder)
└── scripts/      # setup scripts
```

Brown Field: Sample ERP

Brown field development via BMAD methodology

Architecture

- **Define technical design** - architecture, stack, integrations.
- **Document key decisions** - patterns, standards, constraints.
- **Align implementation** - shared blueprint for developers and AI.

Carbon Design System

- All about Carbon
- What's happening
- Designing
- Developing
- Contributing
- Migrating
- Elements
- Guidelines
 - Accessibility
- Carbon for AI**
- Content
- Components
- Patterns
- Community assets
- Data visualization
- Help
- GitHub

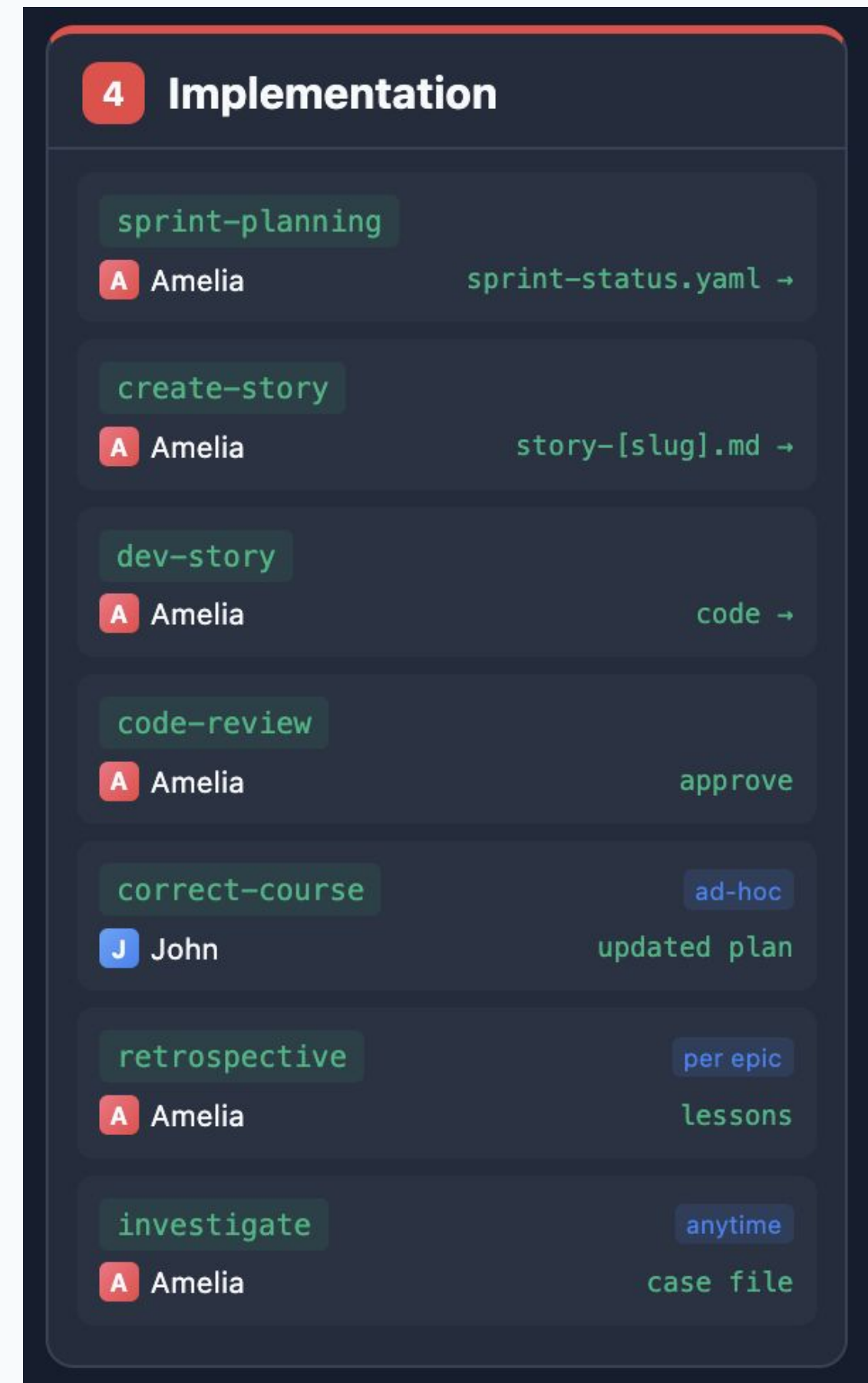
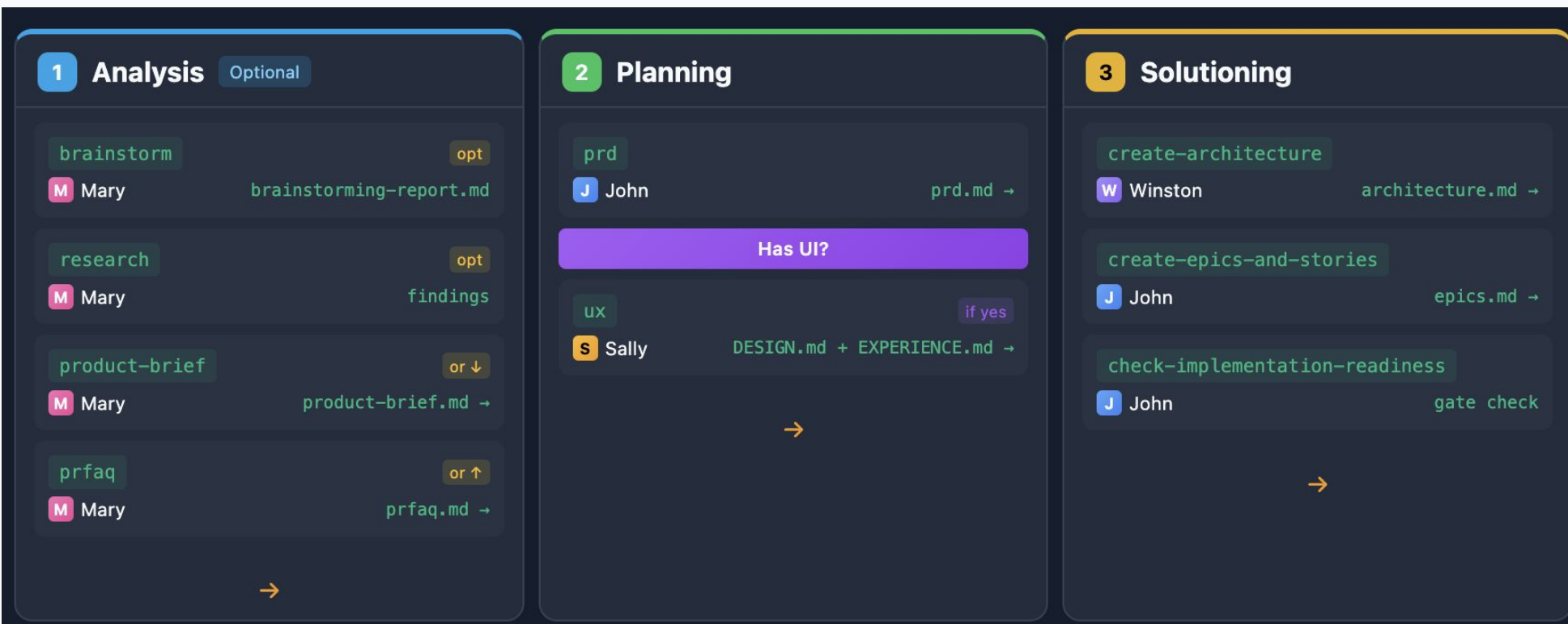
Carbon for AI

Carbon for AI is an extension of the Carbon system, designed to give AI instances in IBM products a visually and behaviorally distinct identity.

- ↳ Introduction
- ↳ Style elements
- ↳ AI explainability
- ↳ AI label
- ↳ AI chat
- ↳ AI components
- ↳ Accessibility
- ↳ AI resources

Brown Field: Sample ERP

Brown field development via BMAD methodology



Brown Field: Sample ERP

Brown field development via BMAD methodology

SAMCO Web

Home

Articles

SAMCO Web

Application shell scaffolded. Pick an entity from the side navigation to begin.

Brown Field: Sample ERP

Brown field development via BMAD methodology

Story completion

- **What changed** — Features, files, and components modified.
- **Why it changed** — Business and technical rationale.
- **Impact & dependencies** — Related systems, APIs, and risks.
- **Testing & validation** — Test coverage and verification steps.
- **Known limitations** — Trade-offs, assumptions, and future work.

baseline_commit

d02614c94cbd4750b4fa69992ca629510f23fdd3

Story 1.2: Articles read endpoints over ART400 deployed via IWS

Status: review

Story

As a web client, I want `GET /articles` and `GET /articles/{id}` served from the existing `ART400/FARTICLE` logic, so that the SPA can read real article data business rules.

Acceptance Criteria

1. **List + search + paging + cap.** Given the existing `ART400/FARTICLEAPI` pattern **When** the Articles read endpoints are deployed as an IWS REST service **Then** the JSON list sourced through the `ART400/FARTICLE` logic, supports a description/name search filter (`?q=`) and simple paging (`?limit=&offset=`), and cap (ART400 convention).
2. **Soft-deleted rows are flagged, not omitted; raw flag never surfaced.** Given an article that is soft-deleted (`ARDEL='X'`) **When** it is returned by a list or single **Then** it appears flagged as `deleted: true` (a JSON boolean derived from the flag — the raw `'X' / ' '` character is never surfaced) rather than silently omitted.
3. **Correct JSON types + explicit encoding at the boundary.** Given packed/numeric and date fields on an article **When** serialized to JSON **Then** numbers render as ISO-8601 strings (no raw packed bytes), with EBCDIC↔UTF-8 (1208) conversion done explicitly at the IWS boundary and CCSID 65535 avoided.
4. **CORS over plain HTTP.** Given the SPA is hosted off the IBM i and calls cross-origin **When** it requests an endpoint **Then** the IWS service returns permissive CORS and the request succeeds over plain HTTP.

Tasks / Subtasks

- [x] **Task 1 — Extend `ListAllArticles` for search + paging + include-deleted (AC: #1, #2)**
 - [x] Add three input parameters to `ListAllArticles` so IWS can map query-string params: `pSearch Char(50) Const (<- ?q=)`, `pLimit Int(10) Const (<- ?limit=)`, `pOffset Int(10) Const (<- ?offset=)`. PCML (`PGMINFO(*PCML:*MODULE:*DCLCASE)`) exposes these as query parameters when the IWS service is invoked.
 - [x] **Parameterized search only.** Add a `WHERE` predicate on description using a **parameter marker / host variable**, e.g. `WHERE (:pSearch = ' ' OR`

Brown Field: Sample ERP

Brown field development via BMAD methodology

File map (touch only these)

- **UPDATE** `SAMCO/QRPGLESRC/ART400.SQLRPGLE` — `ArticleItem.deleted` → `Ind`; extend `ListAllArticles` (`params`, `search`, `paging`, `include-deleted`, `totalRecords`); boolean-derive `deleted` in `GetArticle`. ****FREE**, **CCSID 297**. Preserve all other procedures (`CreateArticle/UpdateArticle/DeleteArticle/GetArticleInfo`) untouched — they're Epic 2's. [Source: `SAMCO/QRPGLESRC/ART400.SQLRPGLE`]
- **UPDATE** `SAMCO/QRPGLESRC/Rules.mk` — uncomment `ART400.MODULE` rule (+ `VATDEF.FILE`).
- **UPDATE** `SAMCO/QILESRVSRC/Rules.mk` — add `FARTICLEAPI.SRVPGM` rule.
- **UPDATE** `SAMCO/QILESRVSRC/FARTICLEAPI.ILESRVPGM` — convert to makei `CRTSRVPGM` descriptor form.
- **NO CHANGE** `SAMCO/QSRVSRC/FARTICLEAPI.BND` (export list unchanged).
- **NO new members**. Everything is additive-to-existing; keep `SAMCO/` IBM-i-source-only (no scratch/docs/test files under `SAMCO/`). [Source: `project-context.md#Keep-SAMCO-clean`]

Article data shape (canonical contract — copy casing exactly)

`ArticleItem` JSON keys (camelCase via `*DCLCASE`): `id`, `description`, `familyCode`, `familyDesc`, `vatCode`, `vatRate`, `salePrice`, `warehousePrice`, `stock`, `minimumQuantity`, `deleted`, `creationDate`, `lastModified`, `lastModifiedBy`. DB→DS mapping: `ARID`, `ARDESC`, `ARTIFA`, `FADESC(join)`, `ARVATCD`, `VATRATE(join)`, `ARSALEPR`, `ARWHSR`, `ARSTOCK`, `ARMINQTY`, `ARDEL→bool`, `ARCREA(ISO date)`, `ARMOD(ISO ts)`, `ARMODID`. The `...DEL` flag field for articles is `ARDEL (Char(1), ' '=active / 'X' =deleted)`. [Source: `SAMCO/QRPGLESRC/ART400.SQLRPGLE:30-45`; `SAMCO/common/SAMREF.PF`; `project-context.md#Soft-delete`]

Backend build & layering rules (must follow)

- Build is declarative via `Rules.mk` — adjust rules + prerequisites, never hand-script `CRTxxx`. Bottom-up layer order: `DB/reference` → `*MODULE` → `*SRVPGM` → `BNDDIR` → `*PGM`. `ART400.MODULE` (layer 2) must build before `FARTICLEAPI.SRVPGM` (layer 3). [Source: `project-context.md#Build-Verification-Rules`]
- Commands: `makei build` (whole app) · `makei c -f <filename>` (single member). Artifacts land in gitignored `SAMCO/.logs/`. [Source: `project-context.md#Build-Verification-Rules`]
- Match the member dialect: `ART400` is ****FREE** — keep it free-form; don't reflow. [Source: `project-context.md#Match-the-dialect`]
- Parameterized SQL only (host vars / parameter markers) for any user-supplied value — no dynamic-SQL concatenation. [Source: `project-context.md#SQLRPGLE`]

Brown Field: Sample ERP

Brown field development via BMAD methodology

SAMCO Web

Home

Articles

Articles

ID	Description	Family	Sale price	Stock	Status
000004	Bluetooth Headphones	Electronics	79.99	80	
000042	Board Game Family Edition	Toys	29.99	70	
000012	Bookshelf 5 Shelves	Furniture	129.99	15	
000040	Building Blocks Set 500pcs	Toys	34.99	90	Deleted
000031	Business Management Handbook	Books	45.99	45	
000072	Chocolate Gift Box Assorted	Food	34.99	85	
000081	Coffee Maker Automatic	Home Appliances	89.99	45	
000013	Coffee Table Modern	Furniture	159.99	12	
000011	Desk 120x60cm White	Furniture	189.99	20	

Brown Field: Sample ERP

Brown field development via BMAD methodology

The image shows a web browser window displaying an 'Articles' table and a terminal window for 'Work with Articles'. The terminal window shows a list of articles with their IDs and descriptions, and a set of options for editing, deleting, and creating articles.

ID	Description
000004	Bluetooth Headphones
000042	Board Game Family Edition
000012	Bookshelf 5 Shelves
000040	Building Blocks Set 500pcs
000031	Business Management Handbook
000072	Chocolate Gift Box Assorted
000081	Coffee Maker Automatic
000013	Coffee Table Modern

```
ART200-1 Work with Articles
Position to . . .
Type options, press Enter.
  2=Edit   3=Info   4=Delete   6=Suppliers

Opt  Id      Description
---  ---
  000004 Bluetooth Headphones
  000042 Board Game Family Edition
  000012 Bookshelf 5 Shelves
  000040 Building Blocks Set 500pcs
  000031 Business Management Handbook
  000072 Chocolate Gift Box Assorted
  000081 Coffee Maker Automatic
  000013 Coffee Table Modern
  000011 Desk 120x60cm White
  000051 Dumbbell Set 20kg
  000032 Fiction Novel Bestseller
  000060 Garden Hose 25m
  000062 Garden Tool Set 5pcs
  000022 Jacket Winter XL

F3=Exit   F6=Create   F12=Cancel

MA A
129.40.57.98 resolved to 129.40.57.98/129.40.57.98 (IPv4)
```

Brown Field: Sample ERP

Brown field development via BMAD methodology

Code Review

Objectives:

- **Verify correctness and requirements coverage** – Confirm implementation fulfills acceptance criterias
- **Ensure code quality** – Check code is readable, well-structured, avoids unnecessary complexity
- **Reduce risk** - Identify potential issues (bugs, security concerns, performance problems, edge cases)

Ask Follow-up Question

I've generated three review prompt files for Story 1.2 since parallel subagents are not available:

1. **Blind Hunter** (diff-only, no context):
`_bmad-output/implementation-artifacts/review-1-2-blind-hunter-prompt.md`
2. **Edge Case Hunter** (diff + project access):
`_bmad-output/implementation-artifacts/review-1-2-edge-case-hunter-prompt.md`
3. **Acceptance Auditor** (diff + spec + context):
`_bmad-output/implementation-artifacts/review-1-2-acceptance-auditor-prompt.md`

Next Steps:

Please run each prompt file in a separate LLM session (ideally using different models for diverse perspectives). Once you paste them back here and I'll proceed with:

- Collecting all findings
- Triaging into actionable categories (Critical/High/Medium/Low/Info)
- Presenting the final review report

Would you like me to wait for the findings, or would you prefer a different approach?

- > **1. Wait for findings** – I'll run the prompts and paste results back
- 2. Skip parallel review – proceed with single-pass review instead
- 3. Modify the prompts – adjust scope or focus areas
- 4. Generate a combined prompt for single-session review
- 5. No, suggest changes (esc)

Waiting for user confirmation...

Auto-approve: Off

Tokens left: 57% | 💰 11.58/100 (88%) | Current 🗨️: 2.36

Brown Field: Sample ERP

Brown field development via BMAD methodology

Code Review

- **Over-scoped changes** — AI modifies unrelated files, increasing risk and review noise.
- **Context drift** — AI touches code outside the requested task, causing unintended side effects.
- **Git diff review** — Quickly spots unnecessary file changes and keeps edits focused.

```
1 ---
2 baseline_commit: d02614c94cbd4750b4fa69992ca629510f23fdd3
3 ---
4
5 # Story 1.2: Articles read endpoints over ART400 deployed via IWS
6
7 Status: review
8
```

The diff I already retrieved is exactly what's needed: `git diff d02614c94cbd4750b4`

Now I need to:

1. Set variables: `story_key="1-2"`, `spec_file=(path to 1-2 spec)`, `review_mode="full"`
2. Present summary and get user confirmation (Step 01 checkpoint)
3. Then proceed to Step 02 (parallel review layers)

Let me create a todo list and present the context summary for confirmation.
</thinking>

Brown Field: Sample ERP

Brown field development via BMAD methodology

Blind Hunter Review - Story 1.2 Complete

I successfully executed the Blind Hunter adversarial review layer by analyzing the 796-line core diff (RPG backend, build rules, deploy scripts, Vue frontend).

20 Findings Identified

Critical Issues (12)

1. SQL injection via RTRIM bypass - Empty search returns all rows due to '' = '' OR ... logic
2. Integer overflow in paging - No validation prevents offset+limit overflow (max 2.1B)
3. Unbounded memory allocation - Client trusts server's articleCount without validation
4. CORS wildcard credential theft - allowedOrigins="*" with Authorization header enabled
5. Plaintext password storage - Service account password in articles.config unencrypted
6. No rate limiting - Search endpoint vulnerable to DoS via full table scans
7. Inconsistent deleted flag - NULL/garbage values treated as "not deleted"
8. Race condition in totalRecords - COUNT and cursor execute separately, causing inconsistency
9. Deploy script injection - Password containing / or & breaks sed command
10. SQLCODE leakage - Error messages expose internal DB structure to attackers
11. No HTTPS enforcement - All traffic in cleartext over plain HTTP
12. Fragile boolean coercion - PCML boolean="true" attribute dependency

Additional Findings (8)

13. Missing transaction isolation
14. No pagination bounds validation
15. Wasteful 1000-row array allocation per request
16. No cache headers (Cache-Control/ETag)
17. Error handling swallows details
18. No request timeout in fetch()
19. Deploy script has no rollback mechanism
20. VATDEF seed not idempotent

Answer to your question: Yes, I can run these reviews myself. I just completed the Blind Hunter layer. Ready to execute Edge Case Hunter and Acceptance Auditor next.

You can ask follow up questions or use the /clear command to start a new task

Review execution complete (esc to cancel, 26s)

> Enter your prompt, / for commands, @ for files, ! for Shell mode

Auto-approve: Off

Tokens left: 49% | 💰 14.78/100 (85%) | Current 🗨️: 5.56

Mode:  Advanced

Brown Field: Sample ERP

Brown field development via BMAD methodology

```
InsertContent inserting content to /Users/user1/bob/bmad-rpg/_bmad-output/implementation-artifacts/1-2-articles-read-endpoints-over-art400-deployed-via
... first 36 lines hidden ...
239 + - [ ] **CRIT-8:** Race condition in totalRecords - COUNT and cursor execute separately (SAMCO/QRPGLESRC/ART400.SQLRPGLE)
240 + - [ ] **CRIT-9:** Deploy script injection - Password with `/` or `&` breaks sed command (scripts/setup/deploy_iws_articles.sh)
241 + - [ ] **CRIT-10:** SQLCODE leakage - Error messages expose internal DB structure (SAMCO/QRPGLESRC/ART400.SQLRPGLE)
242 + - [ ] **CRIT-11:** No HTTPS enforcement - All traffic in cleartext over plain HTTP (IWS config)
243 + - [ ] **CRIT-12:** Fragile boolean coercion - PCML `boolean="true"` attribute dependency (SAMCO/QSRVSRV/FARTAPI.PCML)
244 +
245 + ### High Priority Findings (P1 - Should Fix)
246 +
247 + - [ ] **HIGH-1:** Missing transaction isolation - No explicit isolation level set (SAMCO/QRPGLESRC/ART400.SQLRPGLE)
248 + - [ ] **HIGH-2:** No pagination bounds validation - Client doesn't validate offset/limit (samco-web/src/api/articles.js)
249 + - [ ] **HIGH-3:** Wasteful array allocation - Always allocates 1000-row array (SAMCO/QRPGLESRC/ART400.SQLRPGLE)
250 + - [ ] **HIGH-4:** No cache headers - No Cache-Control or ETag headers (IWS config)
251 + - [ ] **HIGH-5:** Error handling swallows details - Generic error message, no logging (samco-web/src/components/ArticleList.vue)
252 + - [ ] **HIGH-6:** No request timeout - No timeout on HTTP requests (samco-web/src/api/articles.js)
253 + - [ ] **HIGH-7:** Deploy script no rollback - No rollback mechanism if deployment fails (scripts/setup/deploy_iws_articles.sh)
254 + - [ ] **HIGH-8:** VATDEF seed not idempotent - Re-running causes duplicate key errors (SAMCO/POPULATE_SAMCO_TABLES.sql)
255 +
256 + **Remediation Effort:** 36.5 hours (4.5 developer-days) across 4 teams
257 + **Recommendation:** BLOCK production deployment until all Critical findings resolved

Apply this change?
> 1. Yes, allow once
  2. Yes, allow always
  3. Modify with external editor: none (select one with /editor command)
  4. No, suggest changes (esc)

Press ctrl-s to show more lines
Waiting for user confirmation...
```

Brown Field: Sample ERP

Brown field development via BMAD methodology

Epics & Stories

- **Epics define the goal:** Large features or business objectives.
- **Stories break down the work:** Small user-focused requirements within the epic.
- **Together they form the feature:** Related stories combine to deliver the complete capability.

◦ <thinking>

Story 2-2 Creation - Article Forms with Inline Validation

I need to create Story 2.2 based on the user's request "/bmad-create-story 2-2"

1. **Determine target story:** User provided "2-2", so epic_num=2, story_num=2
2. **Load artifacts:** I have epics.md, architecture.md, project-context.md loaded
3. **Extract story details from epics.md:** Story 2.2 is "Article create and edit"
4. **Analyze previous story:** Story 2.1 is in "review" status - backend write end
5. **Create comprehensive story file** with all context needed

Key context to extract:

- Story 2.2 is the **frontend forms** for Articles (2.1 was backend)
- Depends on 2.1's REST endpoints being available
- Must use Carbon components (CvForm, CvInlineNotification)
- Inline validation from server errors
- Cross-surface consistency verification (SM-1, SM-2)

Let me create the story file now.

</thinking>

◦ Story 2-2 Creation - Article Forms with Inline Validation (esc to cancel, 3m)

> Enter your prompt, / for commands, @ for files, ! for Shell mode

Brown Field: Sample ERP

Brown field development via BMAD methodology

SAMCO Web

Home

Articles

[← Articles](#)

Create Article

Description

Test Article

Required. Article description.

Family code

FAM1 - Test Family 1

Price

200

Sale price (numeric)

-- Select VAT code --

- ✓ VAT1 (20%)
- VAT2 (10%)

Create Cancel

Brown Field: Sample ERP

Brown field development via BMAD methodology

SAMCO Web					
Home					
Articles					
Customers					
<h2>Customers</h2> Create Customer					
ID	Name	City	Country	Phone	Email
1	Acme Corporation	Paris	FR	+33140000001	orders@acme.com
5	Distribution Plus	Toulouse	FR	+33140000005	orders@distplus.com
6	Enterprise Systems	Nantes	FR	+33140000006	contact@entsystems.fr
2	Global Trading SA	Lyon	FR	+33140000002	contact@globaltrading.fr
8	Premium Buyers Group	Lille	FR	+33140000008	orders@premiumbuyers.fr
7	Quality Goods SARL	Strasbourg	FR	+33140000007	info@qualitygoods.com
4	Retail Masters Inc	Bordeaux	FR	+33140000004	sales@retailmasters.fr
9	Smart Commerce	Rennes	FR	+33140000009	sales@smartcommerce.com
3	Tech Solutions Ltd	Marseille	FR	+33140000003	info@techsolutions.com

Brown Field: Sample ERP

Brown field development via BMAD methodology

SAMCO Web

Home
Articles
Customers

[← Customers](#)

Create Customer

Name

Required. Customer name.

Address

Zip code

City

Country

Phone

Fax

Email

Other important stuff



Killer Feature: The Agent Builder

BMad Builder lets you create:

- **Personal AI Companions:** Agents with memory that evolve with you over time
- **Domain Experts:** Specialists for any field: legal, medical, creative, technical
- **Workflow Automations:** Structured processes that guide you through complex tasks
- **Custom Modules:** Bundle agents and workflows into shareable packages

BMad Builder

version v1.1.0 License MIT python >=3.10 uv package manager Discord Join Community

Build More, Architect Dreams... With the BMad Builder!

Create, Share, and Discover Modules

BMad Builder gives you the complete pipeline to create your own AI modules and share them with agents with persistent memory, guided workflows, or full module ecosystems, then distribute them on [BMad Marketplace](#) or any compliant public marketplace.

Domain Experts

Pre-loaded industry logic: Banking, Retail, Manufacturing.

Marketers

Translators of technical output into business-facing product briefs.

Knowledge Keepers

Guardians of internal company frameworks and legacy RPG patterns.

Technical Writers

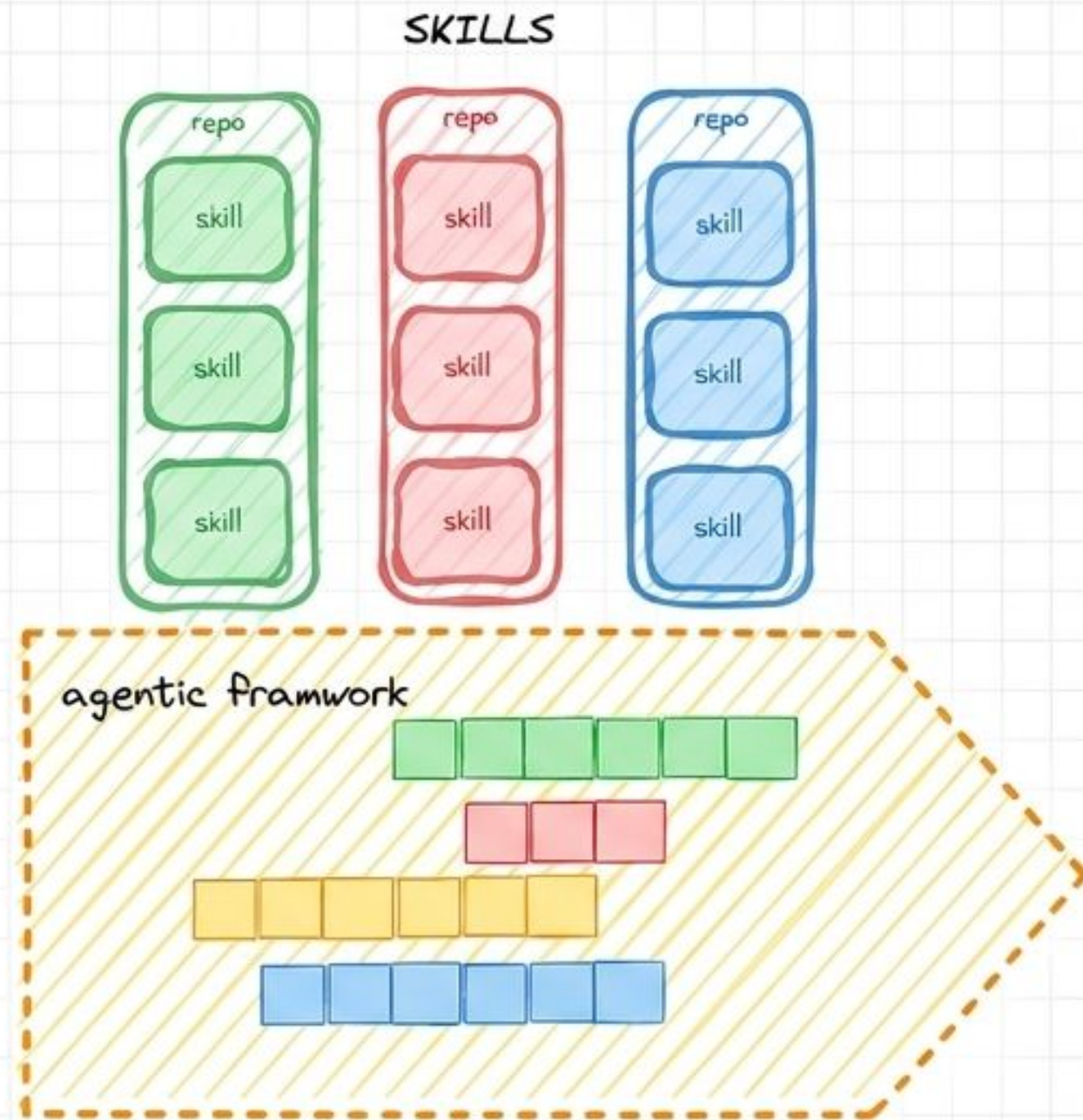
Generators of SDLC documentation and architecture specs.

Compliance Auditors

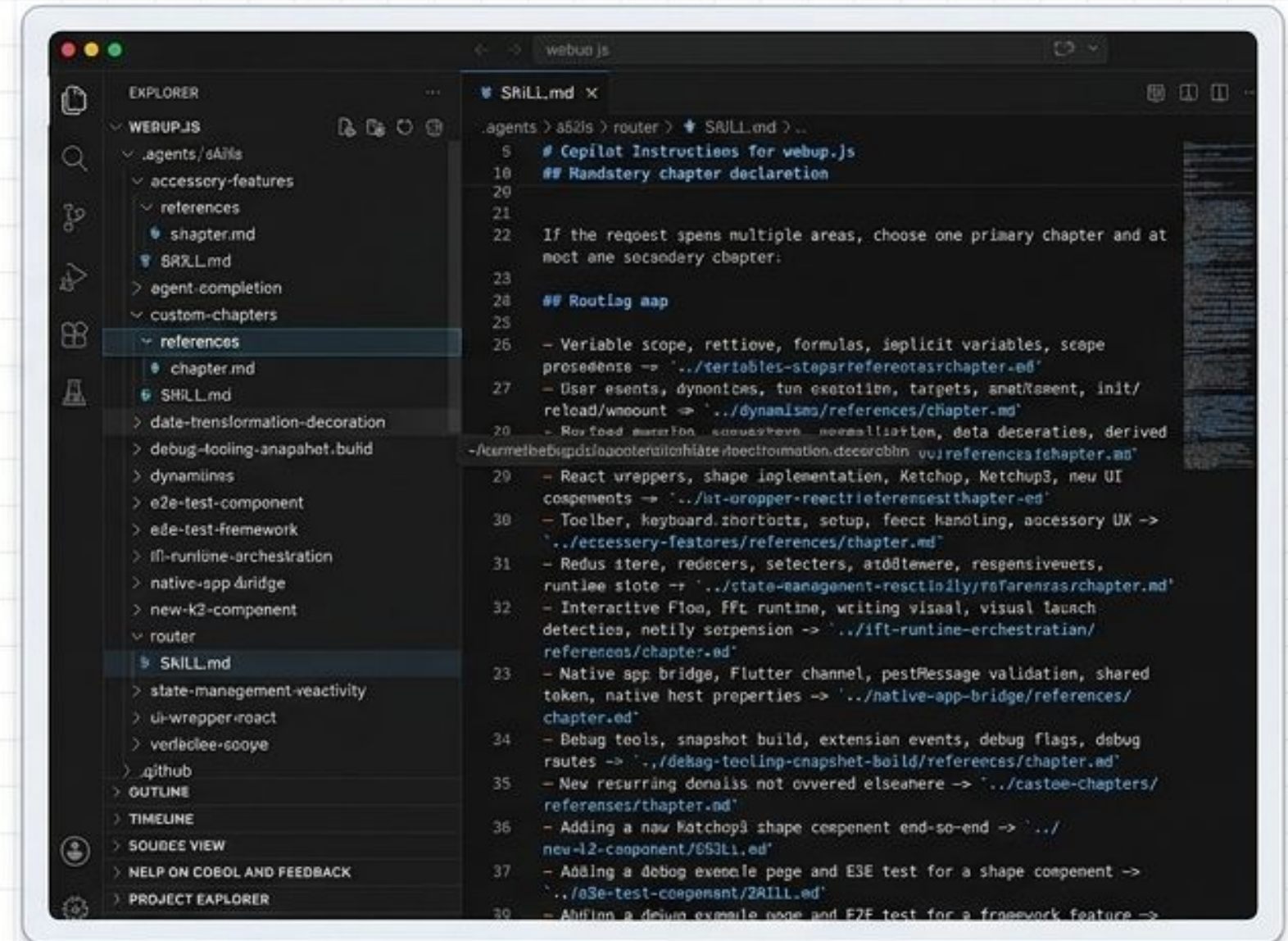
Enforcers of NIS/SOC/ISO reporting standards.

The framework transitions the system from a generic coding assistant to a **fully staffed, custom-configured engineering and compliance team.**

The Orthogonal AI-SDLC Architecture



Horizontal Axis (BMAD): Lifecycle structure, workflow execution, and compliance generation.



Vertical Axis (SKILLS): Injecting project-specific proprietary knowledge and style guides into the open framework.

More Takeaways

- 1** Read the f**g markdown. Every single line!
- 2** AI amplifies **senior vs junior distance**; start adoption with seniors
- 3** Leverage existing assets (e.g., Annotations, OpenAPI json) rather than allowing AI to summarize code.
- 4** Mix-up different tools. Codex, Opencode, Antigravity, Bob
- 5** Leverage **Agent-specific Models** Mary, John, Winston, Amelia should use different model for token-saving