

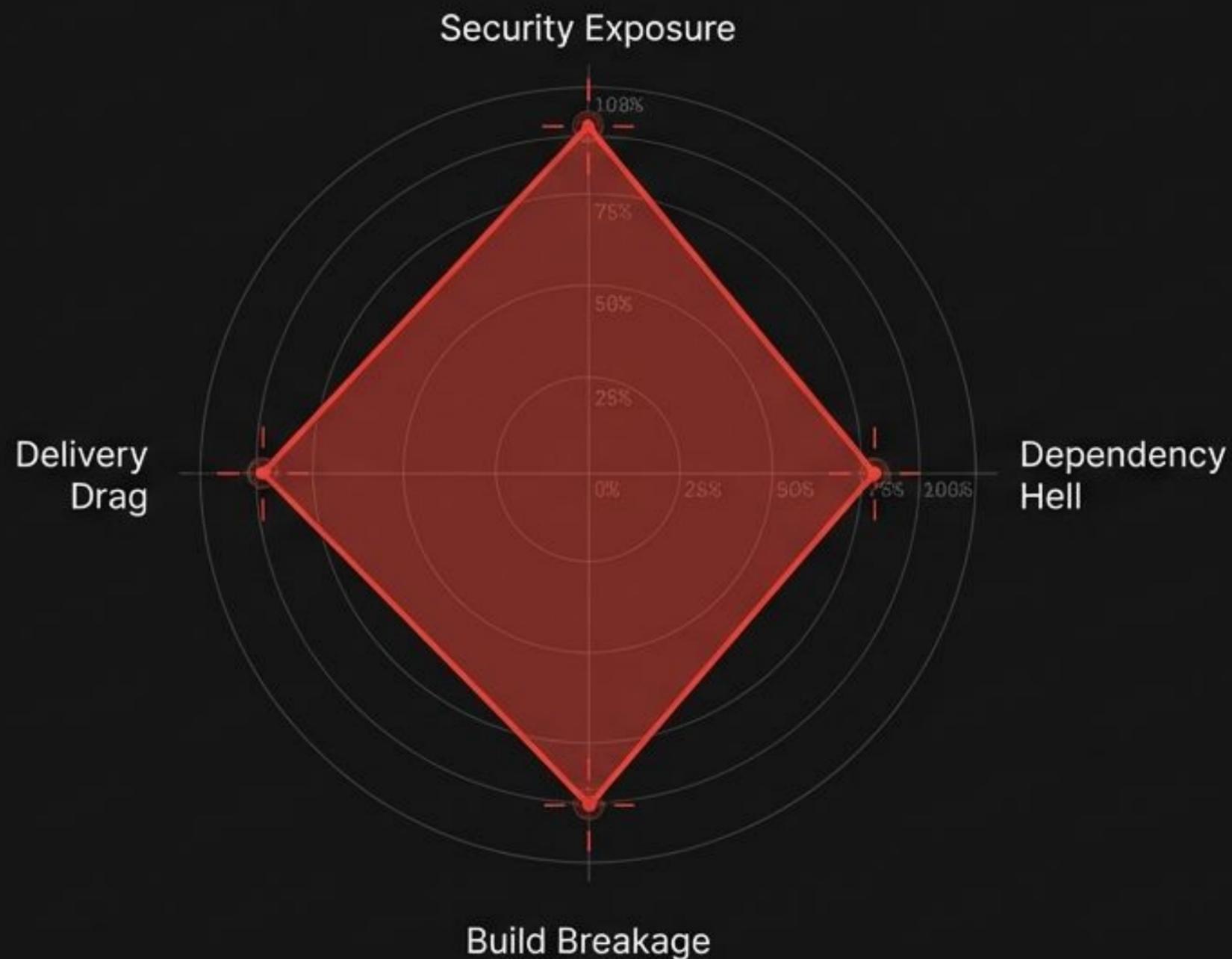
Java Modernization with Gemini CLI & BMaD.

Accelerating the journey from JDK 1.8 to 11 with AI Coding Agents.



RISK RADAR

Current State Analysis: JDK 8



The Legacy Trap: Why We Can't Stay on JDK 8

60%

of enterprise apps are still stuck on Java 8.



Security Exposure: End of support = Open CVEs.



The 'Breakage' List: Removed modules (JAXB, CORBA) break builds.



Dependency Hell: Transitive conflicts block modern libraries.



Delivery Drag: Brittle tests slow down CI/CD.

Structuring the Chaos: The BMaD Framework

A predictable delivery model for AI-driven engineering.

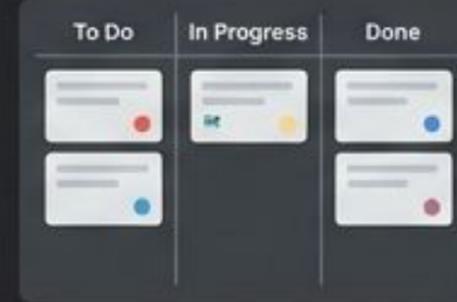
Analysis

Baseline & Debt.



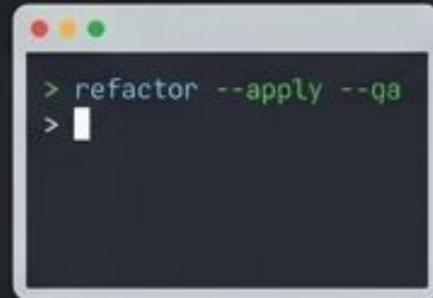
Planning

Sprint Mapping.



Implementation

Refactoring & QA.



Solutioning

Architecture Design.



Meet Your AI-Augmented Squad

Collaborative Intelligence via Personas

Sarah (Product Owner)



> Gemini Task:
Converts BRD to
User Stories.



Winston (Architect)



> Gemini Task:
Scans for Technical
Debt.

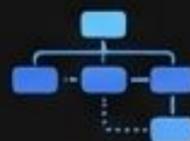


```
vulnerables = {  
  cvulnerable dependency;  
}}
```

Bob (Scrum Master)



> Gemini Task:
Generates Story
Maps.



James (Developer)



> Gemini Task:
Applies Refactoring
Recipes.



```
automated fix {  
  return ~x.refactorer();  
}
```

Quinn (QA Engineer)



> Gemini Task:
Generates Regression
Suites.



The Engine: Gemini CLI Capabilities

```
GEMINI

For getting started:
- Ask questions, edit files, or run commands.
- Use specific prompts for the best results.
- Create GEMINI.md files to customize your interactions with Gemini.
- Visit the docs for more information.

Installation successful! The new version will be used on your next run.

gemini- can you tell what's the current java version of this code repo?

adFile build.gradle
adFile ..adle/wrapper/gradle-wrapper.properties

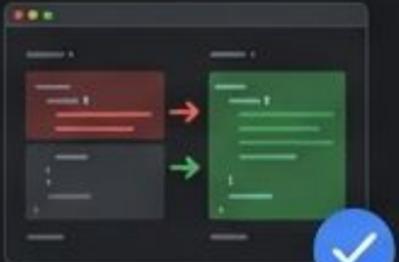
current Java version for this code repository is 1.8. This is specified in the build.gradle file
the line sourceCompatibility = 1.8.

ad:agents:po
acy-java-app (master*) no sandbox (see /docs) auto (99% context left)
```



Whole-Repo Reasoning

Context-aware scanning of dependencies and anti-patterns.



Guided Refactoring

Human-in-the-loop validation of migration recipes.



Test Generation

Automated coverage for modified legacy code.



Phase 2: Planning the Upgrade Sprint

Persona: Bob (Scrum Master)



Readiness Report

```
> /BMad:agents:sm
```

```
↳ Hello, I'm Bob, your Scrum Master. My focus is on preparing detailed, actionable stories for the development team.
```

```
Here are the commands I can execute:
```

1. `help`: Show this list of commands.
2. `correct-course`: Execute the `correct-course.md` task.
3. `draft`: Execute the `create-next-story.md` task to draft a new story.
4. `story-checklist`: Execute the `execute-checklist.md` task with the `story-draft-checklist.md`.
5. `exit`: Exit the Scrum Master persona.

```
I am awaiting your next command.
```

```
> 4
```

```
↳ I can run the story draft checklist for you. Which story would you like me to check? Please provide the story identifier.
```

```
✓ ReadManyFiles Will attempt to read and concatenate files using patterns:...
```

```
### ReadManyFiles Result (Target Dir:  
`/home/rohitbarnwal/gemini_projects/legacy-java-app`)
```

```
Successfully read and concatenated content from **1 file(s)**.
```

```
**Processed Files:**  
- `docs/stories/1.1.story.md`
```

```
> @docs/stories/1.1.story.md
```

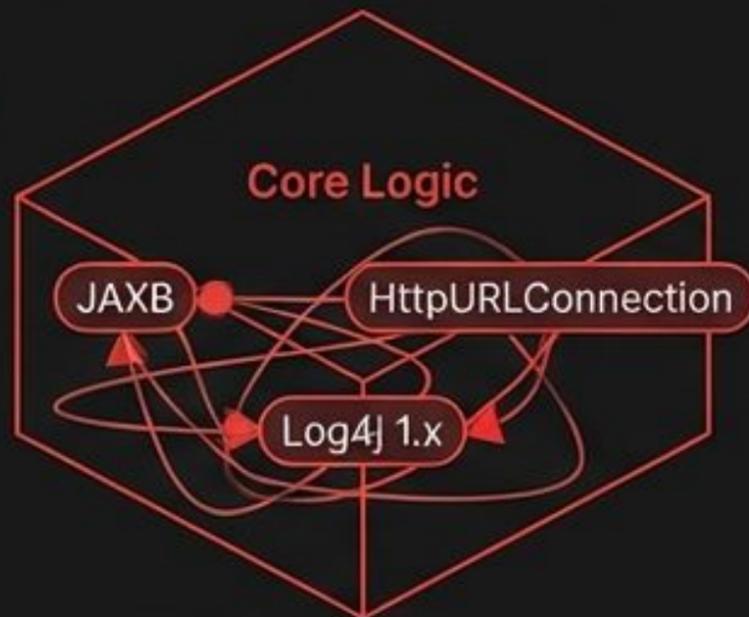
```
↳ Okay, I have the content for Story 1.1. I will now apply the story-draft-checklist.md to it. First, I need to read the checklist itself.
```

```
✓ ReadFile .bm.../.../story-draft-checklist.md
```

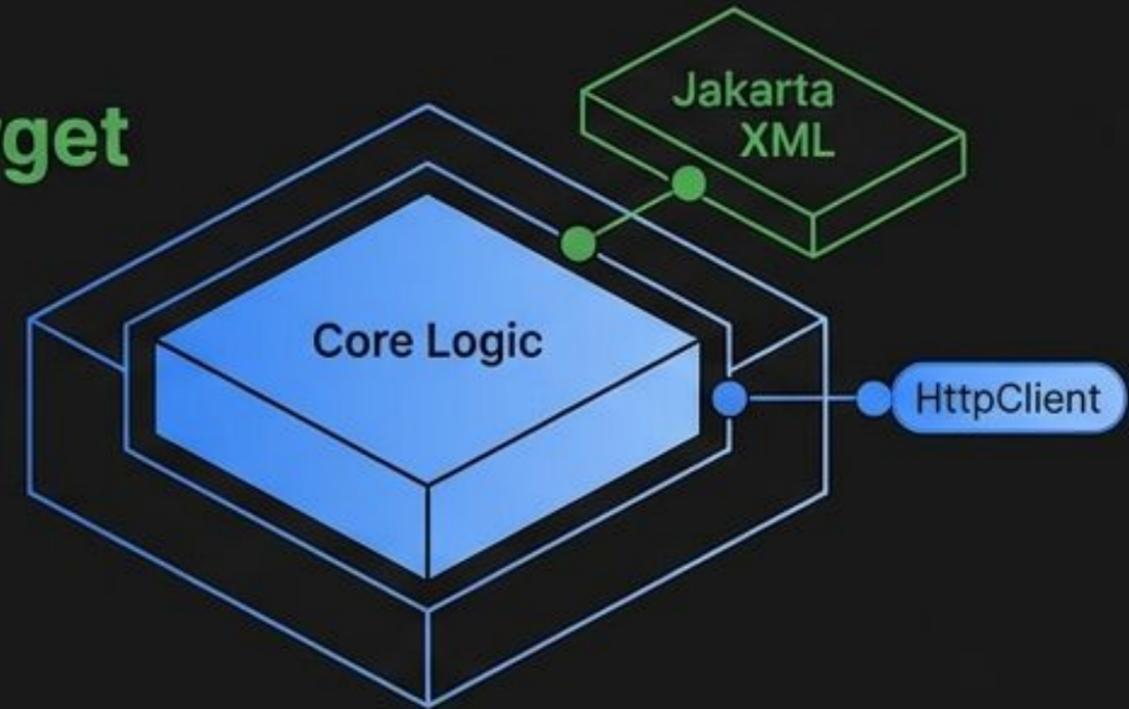
Phase 3: Solutioning & Architecture Blueprint

Persona: Winston (Architect)

Legacy



Target



Component Hierarchy

Core

responsibility: Communication, sending, and output components

interfaces: interfaces: urlc HttpURLConnection

dependencies: Java (JDK 11), Lombok

Data

responsibility: Manage the controls, contexts, and data

interfaces: interfaces: Sensors and Flows

dependencies: Java (JDK 11), Spring Data JPA, H2 Database, Flyway

Presenter

responsibility: Develop the components of the presenter

interfaces: interfaces: Spring Boot Web

tech stack: Java (JDK 11), Spring Boot Web, Spring Security, JWT

Key Decisions

- **JPMS Strategy:** Core modularization vs Automatic Modules.
- **HTTP:** Move to Async HttpClient (HTTP/2).
- **JAXB:** Explicit jakarta.xml.bind dependencies.

\$ `gemini blueprint --input stories.yaml --out architecture.svg`
Generates detailed architectural diagrams.

Phase 4: Implementation via Guided Refactoring.

Persona: James (Developer)

```
TERMINAL

$ gemini refactor --recipe java8to11 --dry-run
Applying recipe...
Modified 5 files.

$ █
```

```
DIFF VIEW

LEGACY (Java 8)                                + TARGET (Java 11)
7  system.out.println(aer, login(), " ");
8  }
9
10 - URL url = new URL(endpoint)
11 - HttpURLConnection con =
    (HttpURLConnection) url.openConnection();
12 - con.setRequestMethod("GET");
13
14 - con.setRequestMethod("GET");
15 - con.setRequestMethod("TOP");
16 // console actions contexts
17 System.out.println(aer, login(), " ");
18 login();
19
20 public static void main(String[] args) {
21 // needs new configuration
22 // ...

```

Workflow Steps:

1. Apply Recipe →
2. Review Diffs →
3. Approve Changes →
4. Fix Build

Code Transformation: Core APIs

HTTP Client

Legacy

```
URLConnection con =  
(URLConnection) new  
URL(  
...).openConnection();  
con.setRequestMethod(  
"GET");
```

Modern

```
HttpClient client =  
HttpClient.newHttpClient(  
);  
var response =  
client.send(req,  
BodyHandlers.ofString());
```

File I/O

Legacy

```
BufferedReader br = new  
BufferedReader(...);  
  
while(line =  
br.readLine())...
```

Modern

```
String content =  
Files.readString(Path.of(  
"config.json"));
```

```
$ gemini transform --apis core --diff-view
```

Code Transformation: Language Features

Collection Factories

Left (Legacy)

```
Arrays.asList("a", "b");  
new HashMap<...>()
```

Right (Modern)

```
List.of("a", "b");  
Map.of("key", value);
```

Gemini

Type Inference (var)

Left (Legacy)

```
HashMap<String, List<Integer>> map = new HashMap<...>()
```

Right (Modern)

```
var map = new HashMap<String, List<Integer>>();
```

Gemini

Type Inference (var)

Left (Legacy)

Right (Modern)

```
HashMap<String, List<Integer>> map = new HashMap<...>()
```

Gemini

Streams (Java 9)

Right (Modern)

```
list.stream().takeWhile(s -> !s.isEmpty())...
```

Gemini

Phase 4: Automated Validation (QA)

Quinn (QA Engineer)

```
$ gemini test --gen --focus changed-files
```



- Test Generation: Filling coverage gaps.
- Regression: Ensuring Legacy Parity.
- Safety: Upgrade JUnit 5 & Mockito.
- Safety: Upgrade JUnit 5 & Mockito.

from

> /BMad:agents:qa

+ Hello, I'm Quinn, your Test Architect & Quality Advisor. I am here to provide comprehensive quality analysis and actionable recommendations.

Here are the commands I can execute:

1. help: Show this list of commands.
2. gate {story}: Execute the quality gate for a story and record the decision.
3. nfr-assess {story}: Validate non-functional requirements for a story.
4. review {story}: Conduct a comprehensive, risk-aware review of a story.
5. risk-profile {story}: Generate a risk assessment matrix for a story.
6. test-design {story}: Create detailed test scenarios for a story.
7. trace {story}: Map story requirements to tests using a Given-When-Then format.
8. exit: Exit the Test Architect persona.

I am ready for your command.

> 4

+ Which story would you like me to review?

> 4

+ Please specify the story identifier you would like me to review (e.g., "1.1", "1.2", or "epic-1-java-17-upgrade").

> epic-1-java-17-upgrade

+ Understood. I will conduct a quality review of the epic epic-1-java-17-upgrade and the surrounding context.

My review process dictates that I should produce a "gate" file summarizing my findings. First, I need to determine the correct location for this file by reading the project's configuration.

✓ **ReadFile** .bmad-core/core-config.yaml

The Modernization Sprint Timeline.

From Analysis to Release in a single flow.



Outcomes: The ROI of AI-Driven Modernization.

60% ↑

Faster Upgrade Cycles
(2.5 Weeks).

72% ✓

AI Code
Acceptance Rate.

< 1.5% ↓

Defect Escape
Rate.

Business Value



Zero CVE Exposure



G1GC Performance



Cloud-Native Ready

Your Path to Modernization.

- 1 Pick a Pilot: **Low-risk service** (e.g., notification-service).
- 2 Run Analysis: **Baseline debt** in Week 1.
- 3 Scale Up: **Templatize recipes** for other services.



Gemini

Possibilities Reimagined with AI.

Outcome: Gemini + BMaD Modernization

- Accelerated transformation cycles
- Complete modernization documentation generated via AI
- Consistent architectural and coding standards
- Fully tested, maintainable, modern application