

# Automate app upgrades & maintenance using Amazon Q Code Transformation

**Vishvesh Sahasrabudhe**

Principal Product Manager  
Amazon Web Services

**Jas Chhabra**

Senior Software Development Manager  
Amazon Web Services

# Agenda

Issues with application maintenance and upgrades

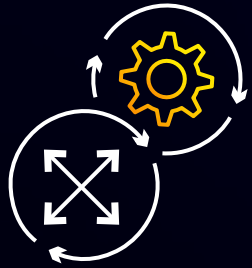
Amazon Q Code Transformation overview

Code Transformation: how it works

Examples of Java upgrades

Future enhancements

# Issues with application maintenance and upgrades



Tedious and time consuming to maintain and upgrade legacy apps



Teams need to upgrade apps from unsupported language versions



Developers want to reduce undifferentiated upgrade tasks



Legacy apps on older language versions pile up and take large dev effort to maintain

# Manual application upgrades



Dependency A version 1.0 -> 2.0

Framework B version 5.6 -> 6.5

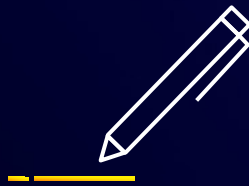
Replace feature X with feature Y

Upgrade test framework

- Version P to version Q



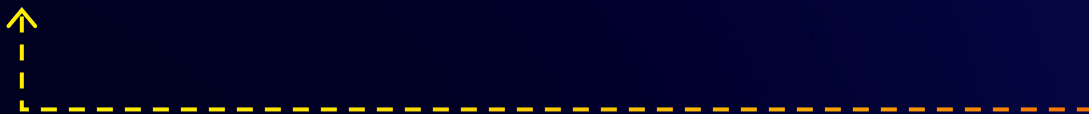
**Research**



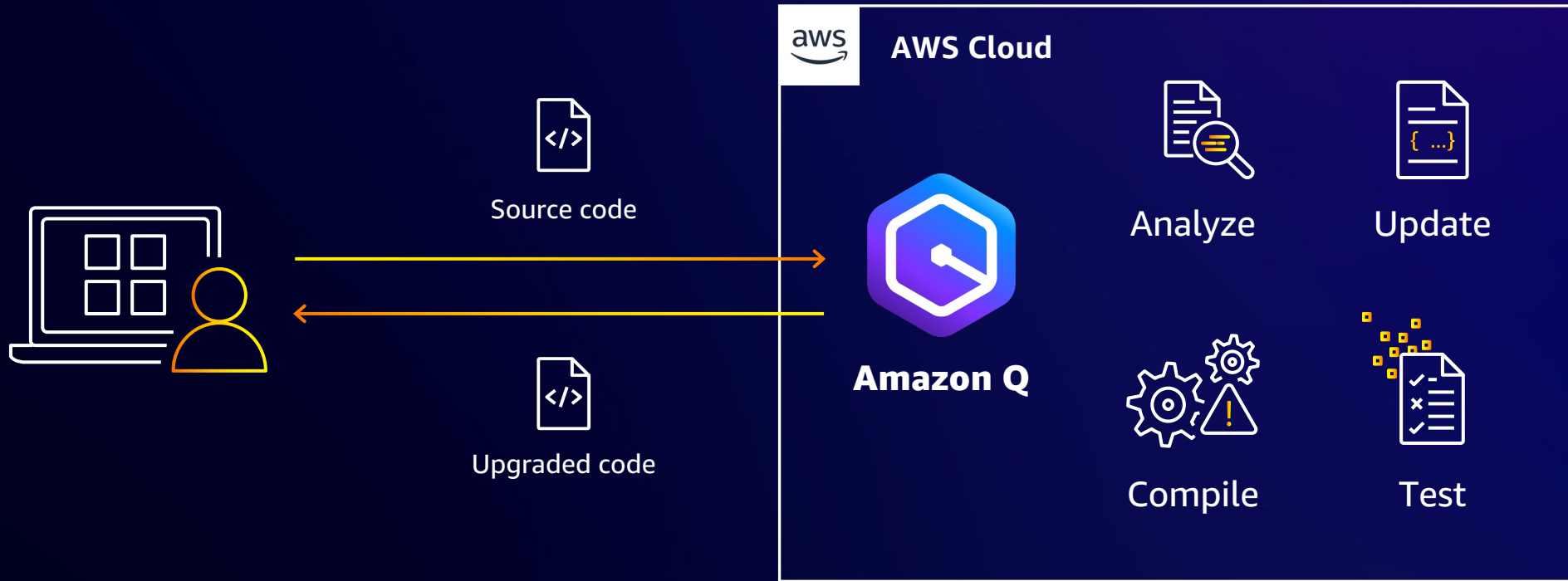
**Write and review**



**Build and test**

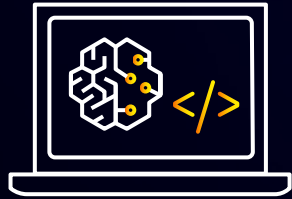


# Introducing Amazon Q Code Transformation



# Transform your application upgrades

KEEP APPLICATIONS SECURE AND RUNNING ON LATEST JAVA RUNTIMES



---

Enhance productivity with  
intelligent, automated  
Java upgrades

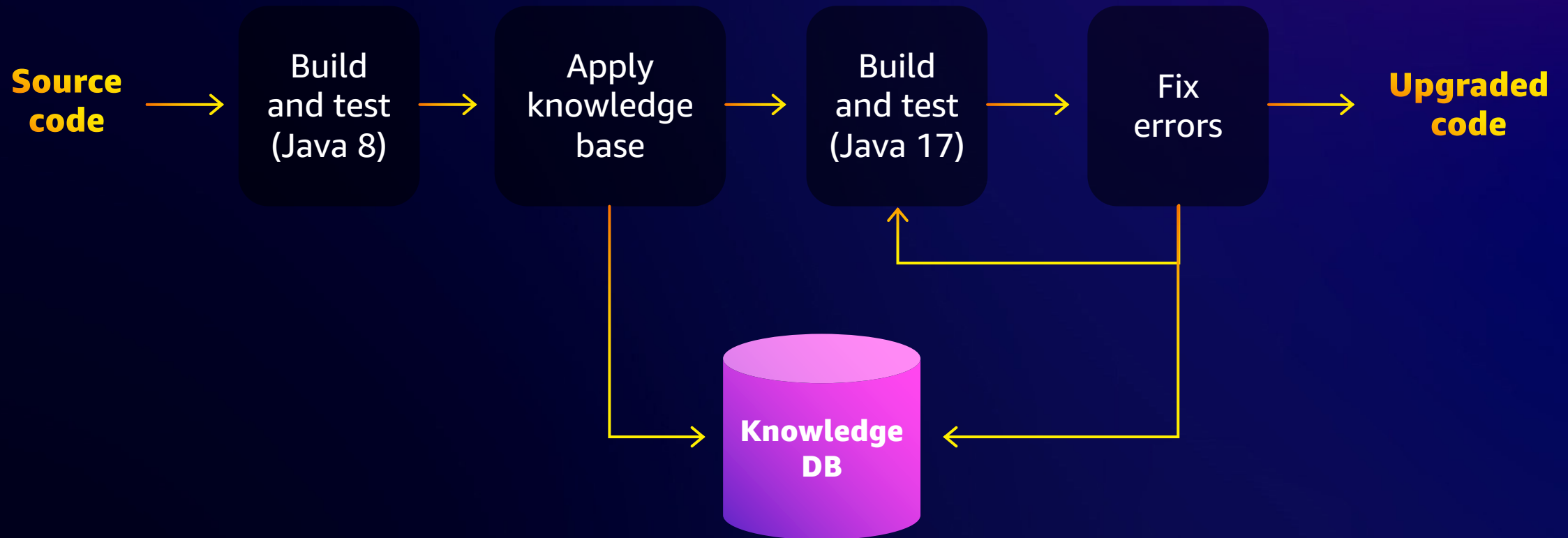


---

Improve security posture  
with apps updated to the  
latest supported version

# Amazon Q Code Transformation: How it works

# Code Transformation: How it works





# Knowledge base

MINING RECOMMENDED VERSIONS OF DEPENDENCIES

```
<dependency>
```

```
<groupId>com.fasterxml.jackson.core</groupId>
```

```
  <artifactId>jackson-core</artifactId>
```

```
-    <version>2.9.4</version>
```

```
+    <version>2.12.5</version>
```

```
</dependency>
```

# Knowledge base

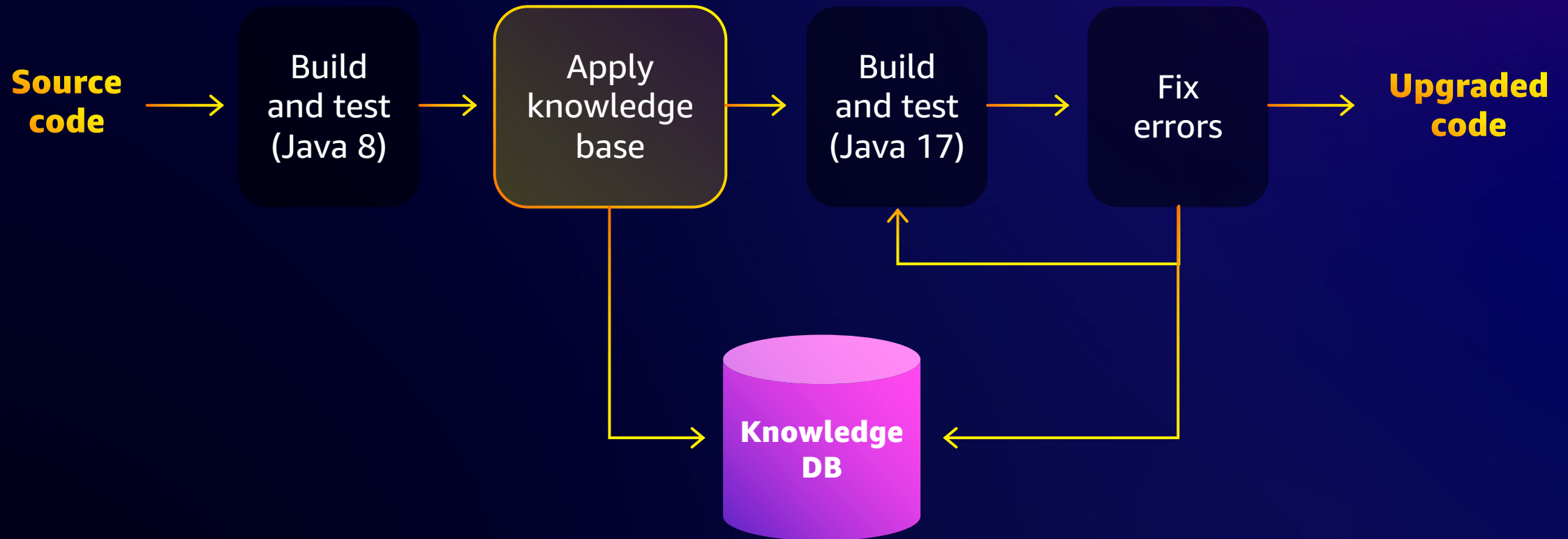
## MINING GENERAL COMPILATION ERRORS AND FIXES

Error: ERROR] [...] package org.junit.runner does not exist

Fix: Add dependency Junit. Example:

```
<dependency>  
  <groupId>junit</groupId>  
  <artifactId>junit</artifactId>  
  <scope>test</scope>  
</dependency>
```

# Code Transformation: How it works



# Applying knowledge base

STATIC TOOLS USED FOR SIMPLE SCENARIOS SUCH AS VERSION UPGRADES

LLMS USED FOR COMPLEX SCENARIOS SUCH CODE CHANGES

```
<dependency>
```

```
<groupId>org.apache.httpcomponents</groupId>
```

```
  <artifactId>httpClient</artifactId>
```

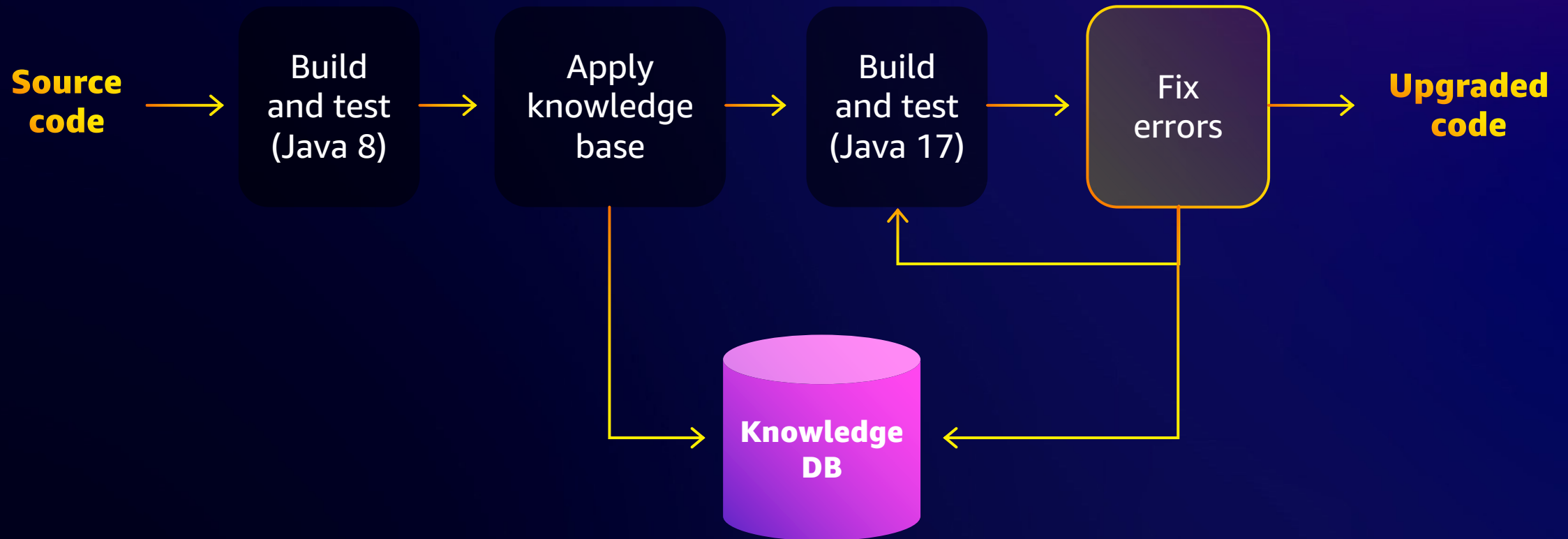
```
-    <version>4.5.5</version>
```

```
+    <version>4.5.13</version>
```

```
</dependency>
```

```
<dependency>
```

# Code Transformation: How it works



# Fixing errors using LLMs

## ERROR RESOLUTION AFTER KNOWLEDGE BASE APPLICATION

### COMPILATION ERROR

src/main/java/com/intuit/benten/flickr/utils/SearchUtils.java:[16,11] exception  
java.io.UnsupportedEncodingException is never thrown in body of corresponding try statement

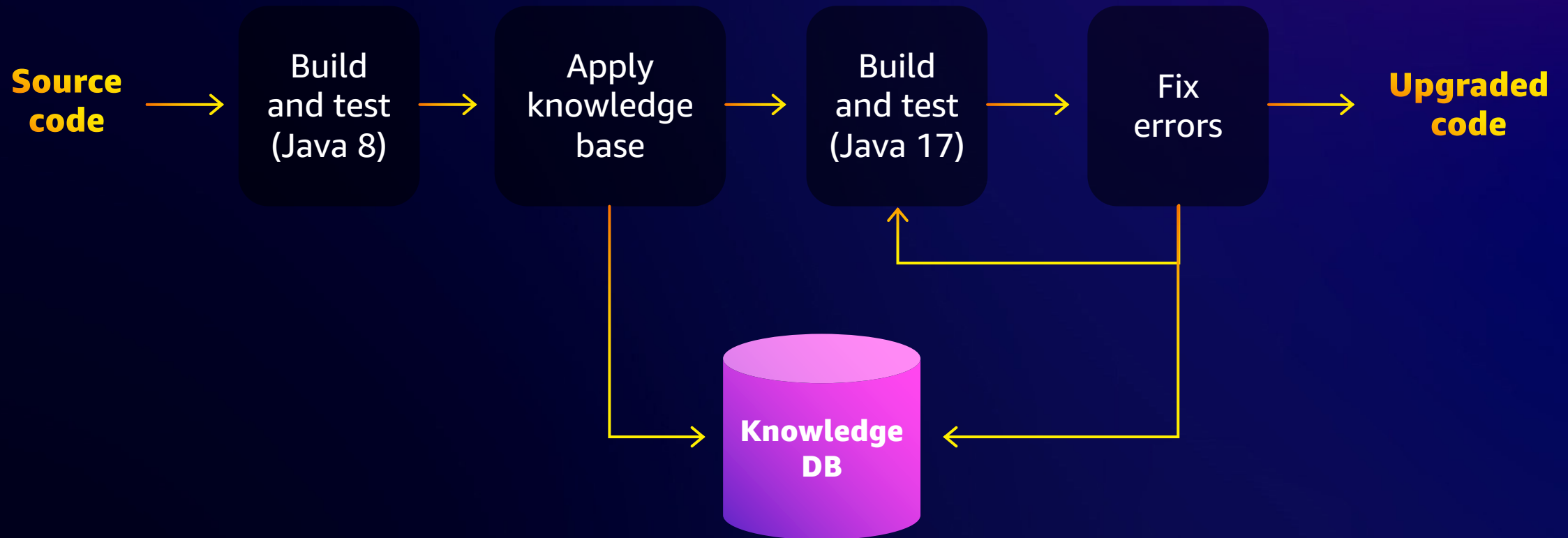
```
public class SearchUtils {  
    public static String generateGoogleSearchUrl(String searchString) {  
        try {  
            String uriQuery = UriUtils.encode(searchString,  
StandardCharsets.UTF_8.name());  
            return SearchItems.GOOGLE_SEARCH_URL + uriQuery;  
        } catch (UnsupportedEncodingException e) {  
            e.printStackTrace();  
            return "";  
        }  
    }  
}
```

# Fixing errors using LLMs

LLM CODE UPDATE THAT FIXES THE ERROR

```
public class SearchUtils {  
    public static String  
    generateGoogleSearchUrl(String searchString) {  
        String uriQuery = UriUtils.encode(searchString,  
            StandardCharsets.UTF_8.name());  
        return SearchItems.GOOGLE_SEARCH_URL + uriQuery;  
    }  
}
```

# Code Transformation: How it works (recap)





# Code Transformation: Demo



# Additional upgrade examples

## COMPILATION ERROR

src/test/java/com/example/demo/controller/HelloControllerTests.java:[22,44] cannot find symbol  
symbol: method secure()

location: @interface org.springframework.boot.test.autoconfigure.web.servlet.WebMvcTest

LLM Fixes the compilation error by replacing

`@WebMvcTest(value = HelloController.class)`

with

`@WebMvcTest(value = HelloController.class, secure = false)`

# Additional upgrade examples

Original compatible with Junit4

```
@RunWith(SpringRunner.class)
@WebMvcTest(value = StudentController.class, secure =
false)
public class StudentControllerTest {
```

Upgraded to Junit5

```
@WebMvcTest(value = StudentController.class)
class StudentControllerTest {
```

# Limitations: First-party upgrades

Code Transformation can upgrade open-source packages

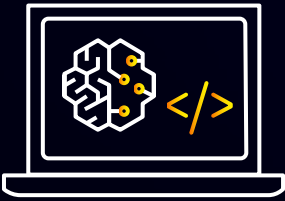
Internal dependencies are not upgraded during the transformation

- We will attempt upgrades of open-source packages in such applications
- You might need to update internal dependencies if they are incompatible with Java 17



# Amazon Q Code Transformation

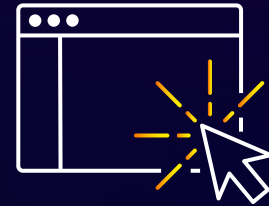
MAINTAIN, UPGRADE, OR MIGRATE YOUR APPLICATION IN JUST A FEW CLICKS



Enhance productivity with intelligent, automated Java upgrades



Improve security and performance with apps updated to the latest supported version



Save time and costs with accelerated Windows to Linux migrations with .NET upgrades\*

A small AWS team upgraded 1000 Java apps from Java 8 to 17 in **just 2 days**, with **10 minutes** on average to upgrade applications, compared to 2 or 3 days spent upgrading each application, earlier

# Thank you!



Please complete the session  
survey in the mobile app