



TEST IO

# Integrating Generative AI for Testing into the Software Lifecycle



Tariq King

CEO and Head, Test IO

VP and Managing Principal, Technology, EPAM

# FLASH BACK



MANUAL SOFTWARE TESTING

PRESENT EXPLORATION AND TESTING



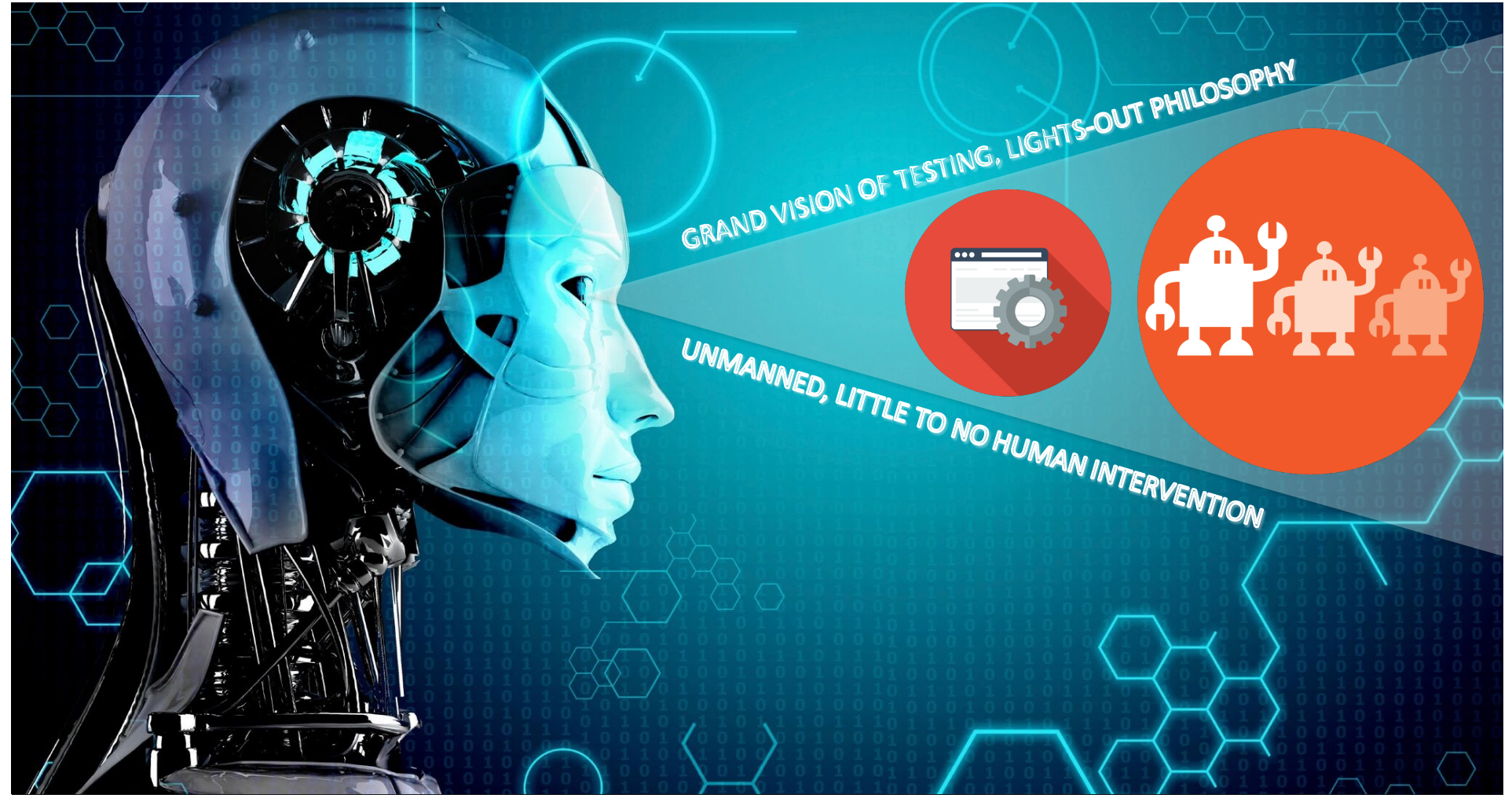
# 2015

TEST SCRIPTS

AND MAINTENANCE

MANUAL



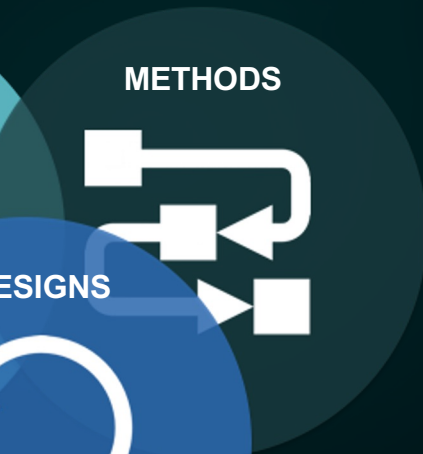
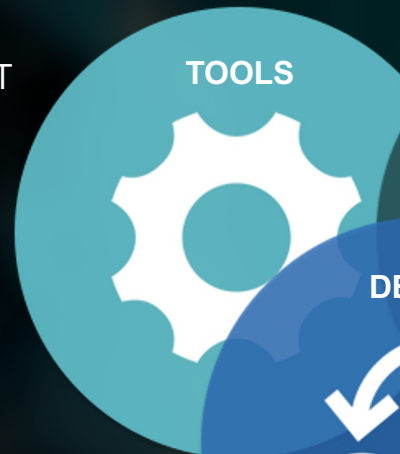


GRAND VISION OF TESTING, LIGHTS-OUT PHILOSOPHY



UNMANNED, LITTLE TO NO HUMAN INTERVENTION

AI-DRIVEN TEST  
AUTOMATION



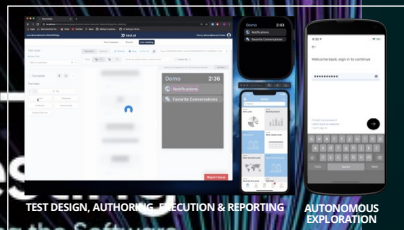
TESTING AI  
SYSTEMS



SELF-TESTING  
SYSTEMS

O'REILLY DESCRIBES PRACTICAL AIST EXPERIENCES

AI  
Test



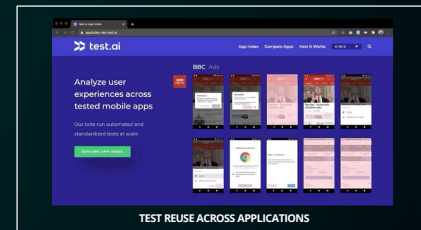
TEST DESIGN, AUTHORING, EXECUTION & REPORTING  
AUTONOMOUS EXPLORATION

Bridging the Software  
Automated AI FOR TESTING WEB, MOBILE AND DESKTOP

Tariq Kin

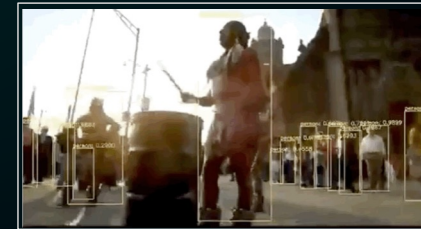


REPORT  
AI FOR TESTING VIDEO GAMES

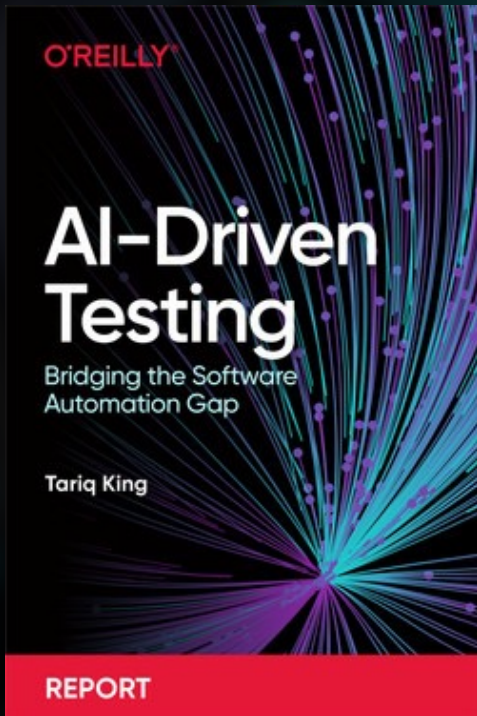


TEST REUSE ACROSS APPLICATIONS

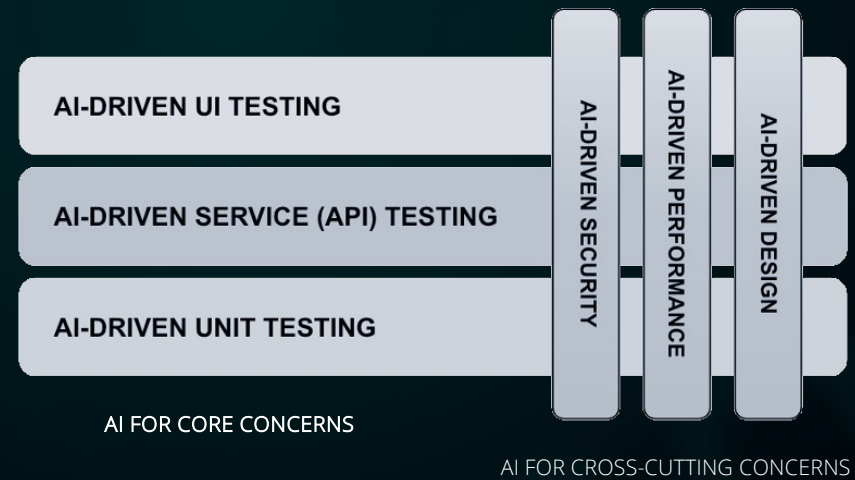
AI FOR TESTING APPSTORES

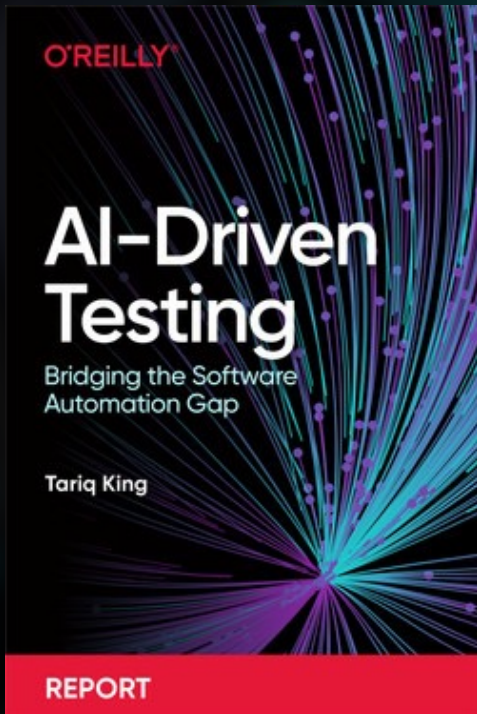


AI FOR TESTING VIDEO QUALITY

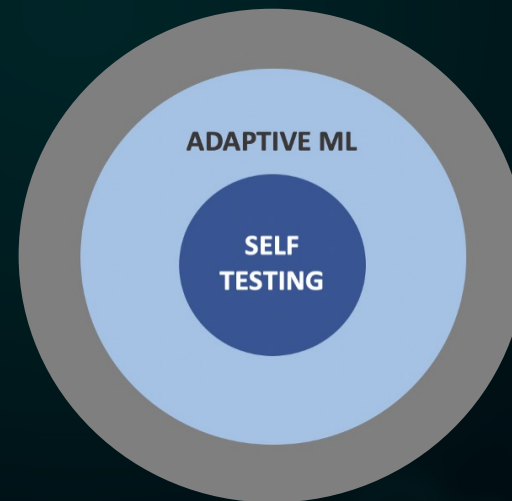


## PREDICTS **FULL-STACK AIST TAKEOVER** AND...





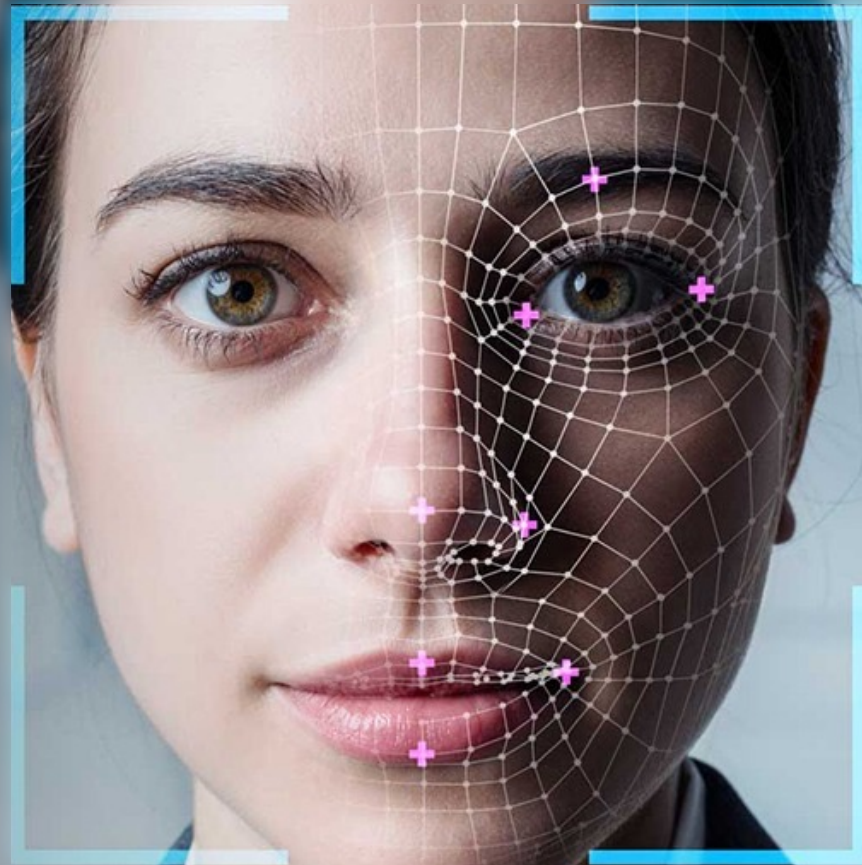
## THE NEED FOR SELF-TESTING AI





# 2024

# GENERATIVE AI



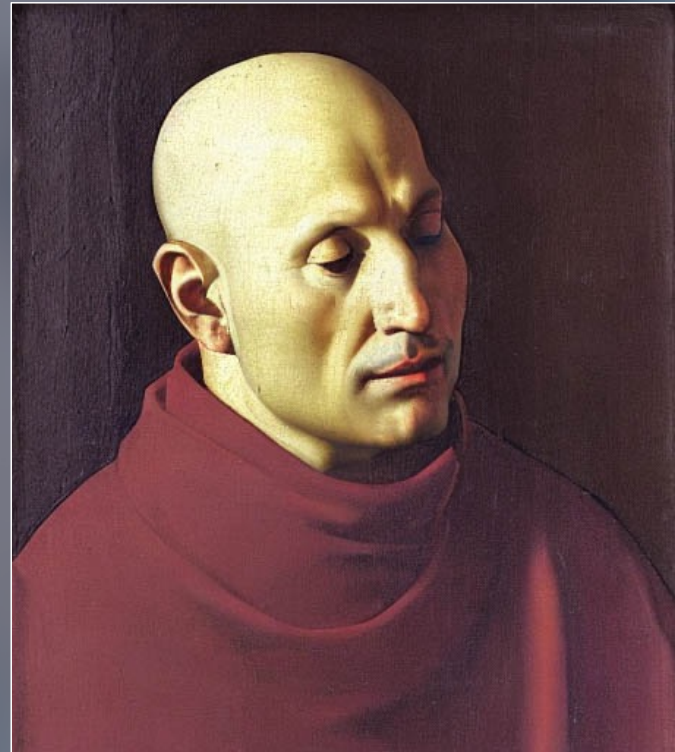
# GENERATIVE AI

Describes computer algorithms that can be used to create natural language, audio, images, and videos with little to no human intervention

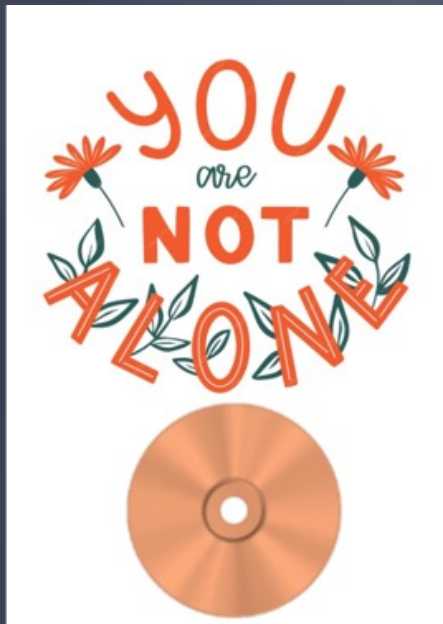
# AI GENERATES PHOTO-REALISTIC FACES



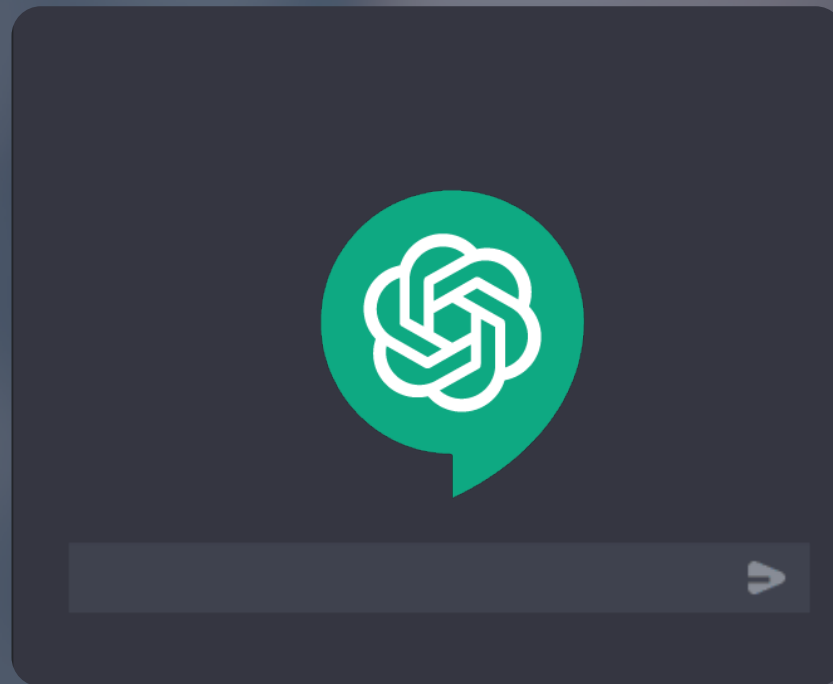
# AI GENERATES ARTWORK



# AI GENERATES MUSIC



# AI GENERATES NATURAL LANGUAGE



# MADE AI BELIEVABLE

USABLE



VISIBLE

COMPREHENSIBLE





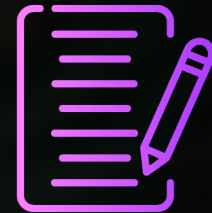
# GEN-AI ASSISTED SOFTWARE TESTING



TEST CASE DESIGN  
AND CREATION



TEST SCRIPT  
GENERATION



TEST PLANNING AND  
DOCUMENTATION



TEST MAINTENANCE  
AND MANAGEMENT




SYNTHETIC TEST  
DATA GENERATION



TEST EXECUTION, DEFECTS  
AND REPORTING





**Lots of tasks are completed**  
(QUANTITY)

The diagram consists of three overlapping circles. The top circle is orange and contains the text 'Lots of tasks are completed (QUANTITY)'. The bottom-left circle is dark blue and contains the text 'Finished work is excellent (QUALITY)'. The bottom-right circle is light blue and contains the text 'Little effort is wasted (EFFICIENCY)'. The central intersection of all three circles is a darker blue and contains the word 'Productivity'. The background features faint illustrations of a robot head on the left, a person with a telescope on the right, and a large upward-pointing arrow.

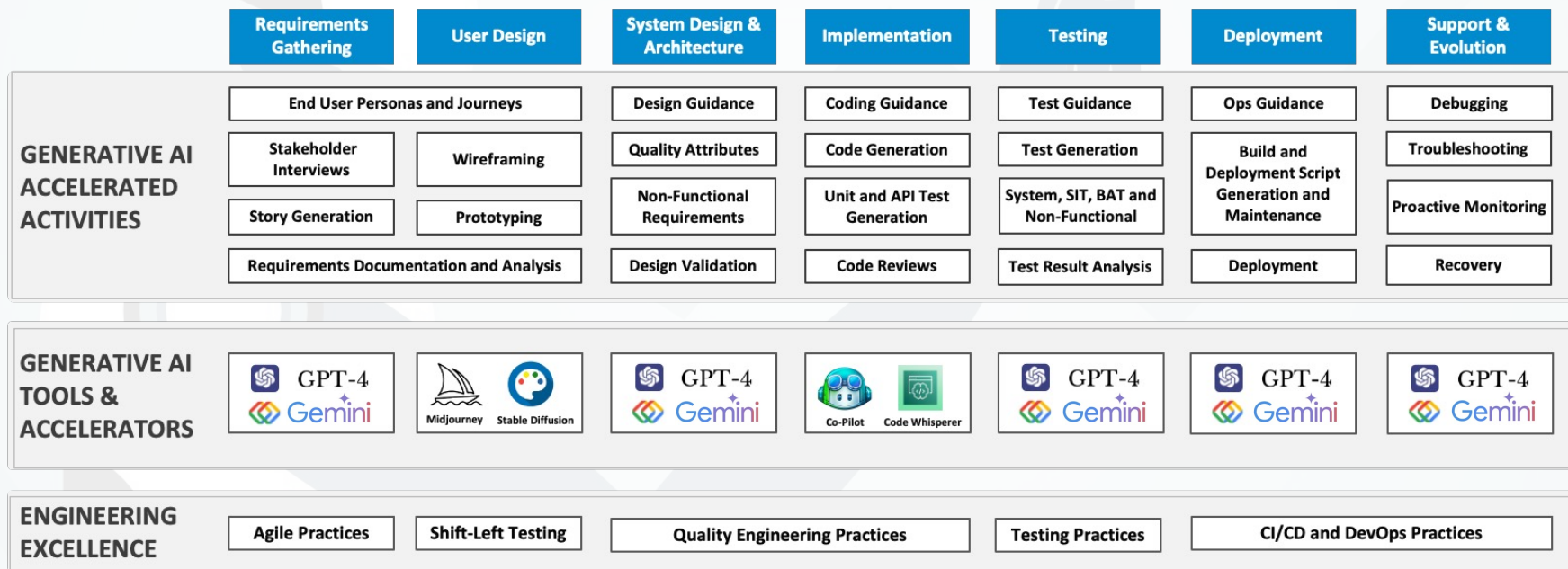
**Productivity**

**Finished work is excellent**  
(QUALITY)

**Little effort is wasted**  
(EFFICIENCY)


# AI ASSISTED SOFTWARE LIFECYCLE

GENERATIVE AI FOUNDATION, TOOLS, ACCELERATORS AND ACTIVITIES



# AI FOR REQUIREMENTS ENGINEERING

GENERATING USER STORIES WITH ACCEPTANCE CRITERIA AND UI MOCKUP DESIGNS

 Agile User Story Generator

▼ **Wishlist Addition**

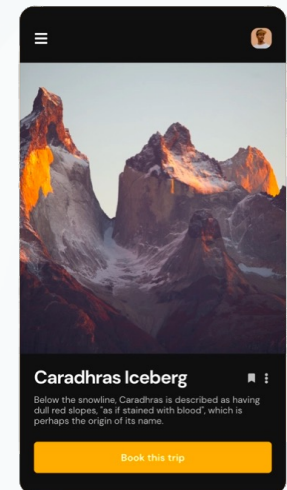
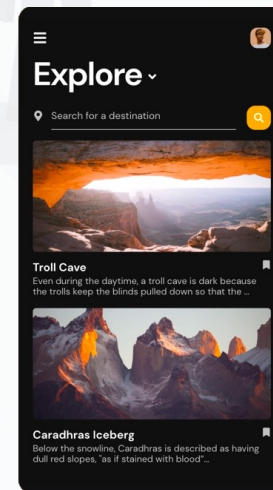
1. As a registered user,
2. I want to be able to add items to my wishlist,
3. So that I can save items I'm interested in for later purchase.

▼ **Acceptance Criteria**

1. Given that I am a registered user, when I view a product, then I should see an "Add to Wishlist" button.
2. Given that I click the "Add to Wishlist" button, the product should be added to my wishlist.
3. Given that a product is in my wishlist, when I visit my wishlist page, then I should see all the products I've added.
4. Given that a product is in my wishlist, when I click on the product, then I should be taken to that product's page.
5. Given that a product is in my wishlist, I should have an option to remove it from my wishlist.

▼ **Notes**

- The "Add to Wishlist" button should not appear for items that are already in the user's wishlist.
- Users should be able to add items to their wishlist from both product listing and product detail pages.
- Users should be able to access their wishlist from the main navigation menu.
- If the product becomes unavailable or out of stock, the user should be notified in the wishlist.



# AI FOR REQUIREMENTS ENGINEERING



AI JEANNIE

**ATLASSIAN** Marketplace



**AI Jeannie**

by **EPAM Systems, Inc.**

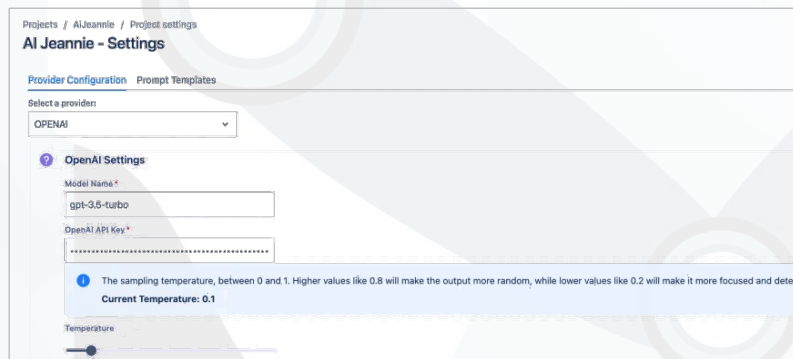
for Jira Cloud

Get it now

Free app

4/4 ★★★★★ 1 ↓ 27 installs

Bring Your Own AI Provider



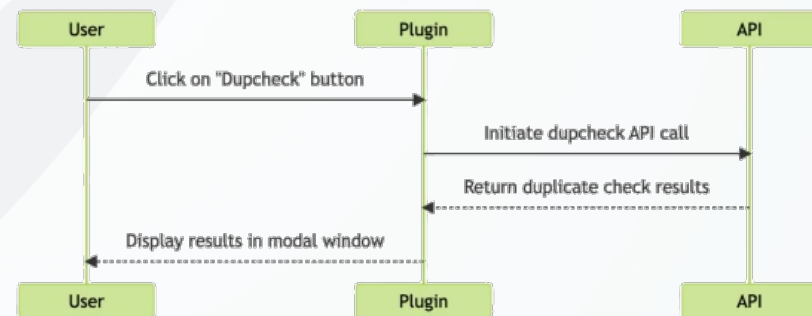
## AI-Powered Epics and User Stories

Generate Acceptance Criteria from Natural Language Prompts

Generate AC

Copy to Description

## Generate Sequence Diagrams for Requirements Analysis



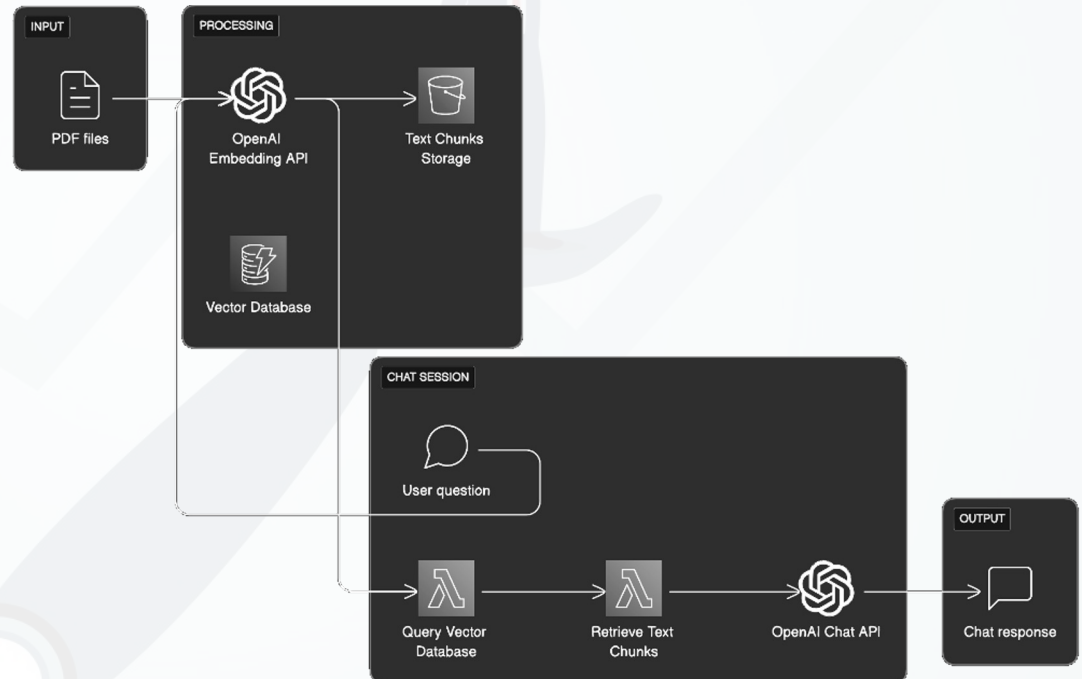
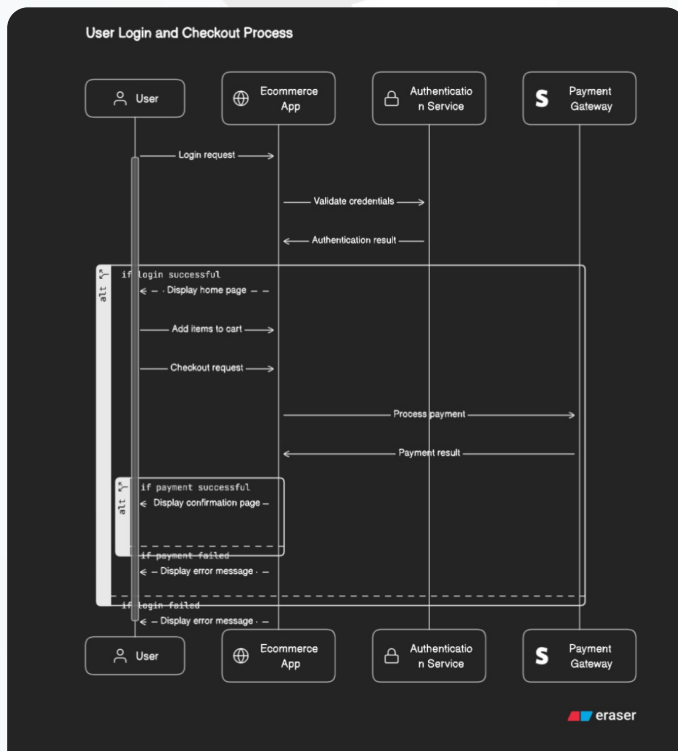




A

# AI FOR SOFTWARE DESIGN

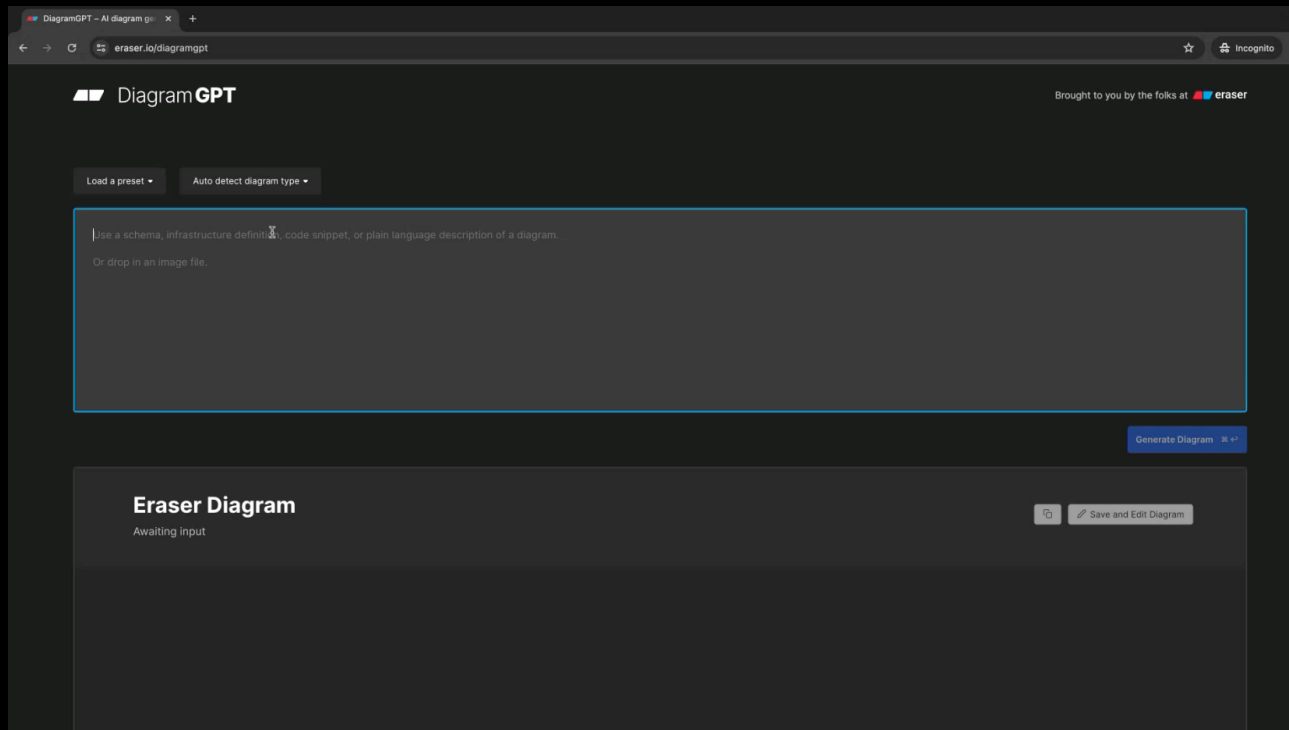
GENERATING DESIGN AND ARCHITECTURE MODELS





Diagram**GPT**

## DiagramGPT in Action!



### Prompts

Generate a cloud architecture for a social media application

- Similar to Instagram
- Design should use AWS components like:
  - EC2 for Compute
  - S3 for Data Storage
  - RDS for database management
- Ensure architecture supports scalability, data privacy and efficient content delivery

# AI FOR SOFTWARE DEVELOPMENT

GENERATING ALGORITHMS AND PROGRAM CODE

```
0 references | 0 changes | 0 authors, 0 changes
39 public static void CreateTables()
40 {
41     using (var context = new TaskContext())
42     {
43         context.Database.ExecuteSqlRaw("CREATE TABLE tasks (id INT PRIMARY KEY, title VARCHAR(50), priority INT)");
44     }
45 }
46
47
48
49
50
51
52
53
54
55
56
57
58
59 }
60
```

# AI FOR SOFTWARE TESTING

## GENERATING AUTOMATED TEST SCRIPTS



```
src > bankService > J CheckingAccount.java > ...
1 package bankService;
2
3 import bankService.exceptions.InsufficientFundsException;
4 import bankService.exceptions.InvalidAmountException;
5 import notificationService.Notification;
6
7 /**
8  * Specialized bank account on which checks can be processed.
9  * @author Tariq King
10  */
11 public class CheckingAccount extends Account {
12
13     private double minBalance;
14     private double overdraftLimit;
15     private double overdraftFee;
16     private double serviceFee;
17     private boolean isOverDrawn;
18     private boolean droppedBelowMinBalance;
19     private boolean notificationsEnabled;
20     private Notification notificationService;
21
22     public CheckingAccount(Customer owner, double startBalance, long accountNumber) {
23         super(owner, startBalance, accountNumber);
24         this.minBalance=1500.0;
25         this.overdraftLimit=0.0;
26         this.overdraftFee=30.0;
27         this.serviceFee=12.0;
28         this.isOverDrawn=false;
29         this.droppedBelowMinBalance =false;
30     }
31 }
```

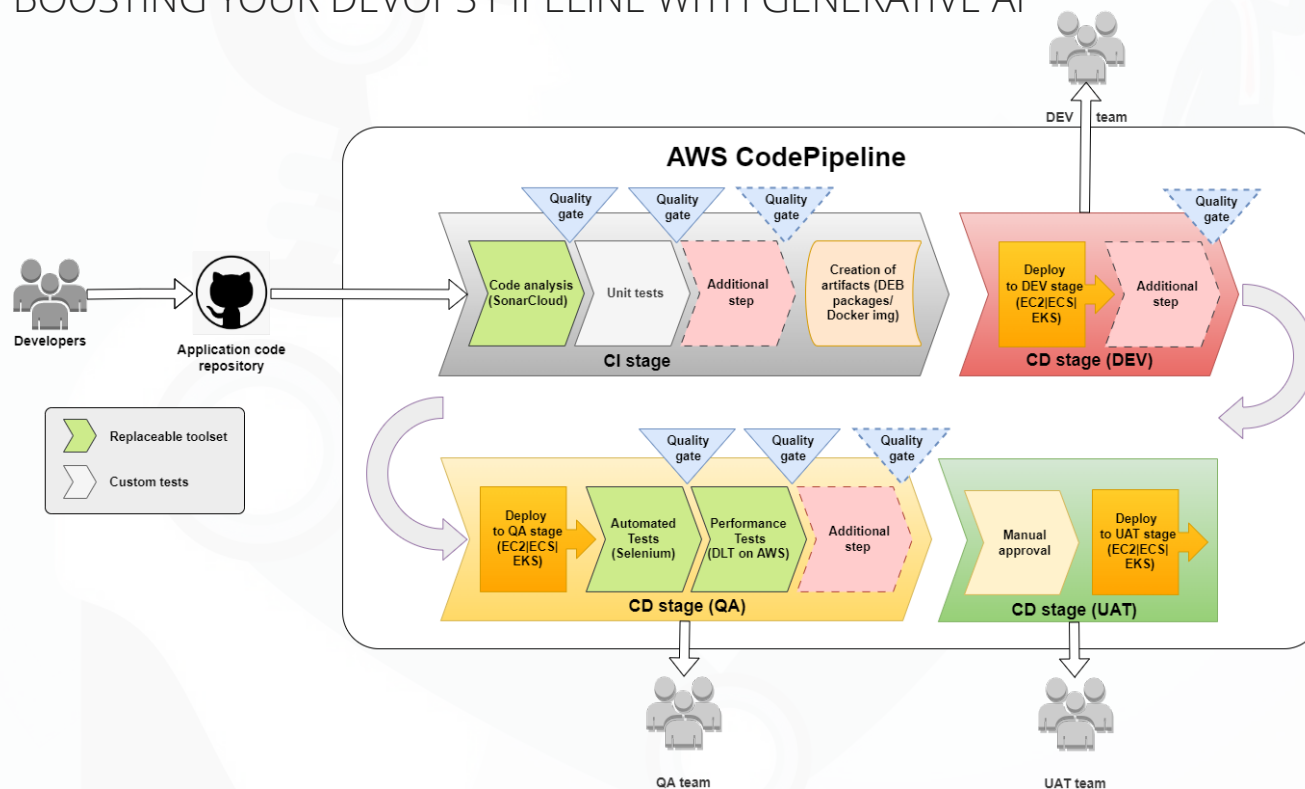
## ELITEA™

### Prompt Summary

- Act as a Java developer
- Understand the code
- Generate unit tests
- Use testing techniques
- Follow best practices
- Here are examples...

# AI FOR CI/CD

BOOSTING YOUR DEVOPS PIPELINE WITH GENERATIVE AI



## AI in Terraform Code:

- Error processing via PR with fixes
- Regula warnings processing via PR comments

## AI in Application Code:

- Unit test generation
- Code explanation
- Code comment composition

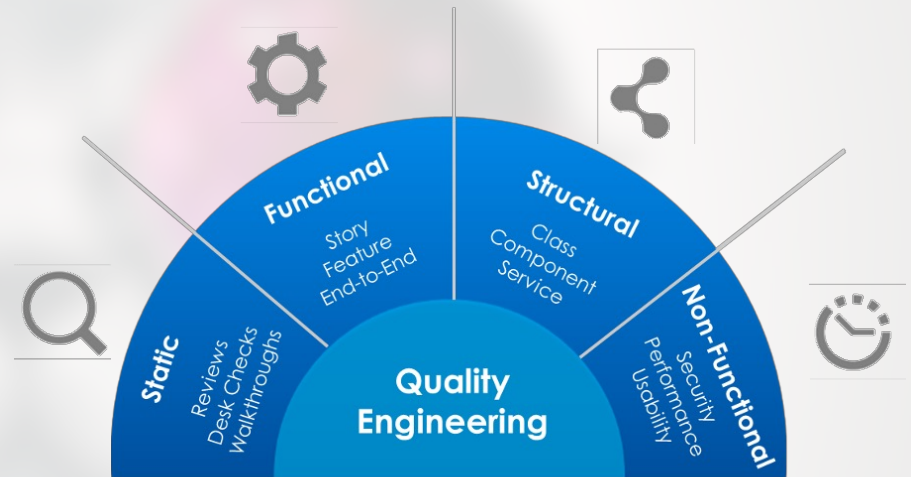
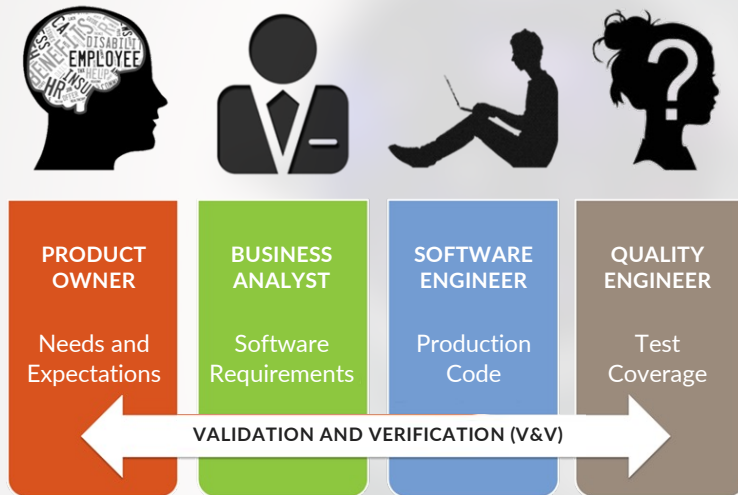
## AI in Quality Gates:

- Issue handling via PR with fixes



# AI IS AUTOMATING QUALITY ENGINEERING

BUILDING QUALITY IN, RATHER THAN TESTING IT OUT AT THE END



# BUT ITSELF RAISES NEW CHALLENGES

## NAVIGATING THE ROAD TO SUCCESSFUL AI ADOPTION



Uncertainties in areas of security and intellectual property ownership when using large language models.



Upskilling is needed to learn how to interact with generative AI tools to use them effectively.



Wrong input results in the wrong answer.  
Correct inputs don't guarantee right answer.

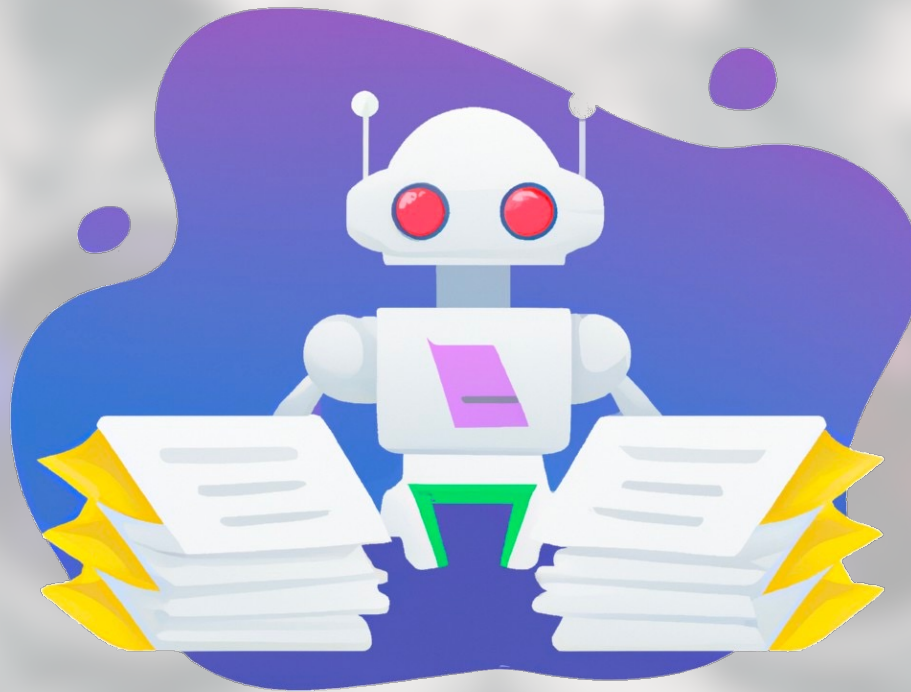


Tools may generate large quantity of software artifacts (code, tests) that look correct to the untrained eye.



Measuring productivity (quality) is non-trivial.  
Faster output doesn't always mean better results.



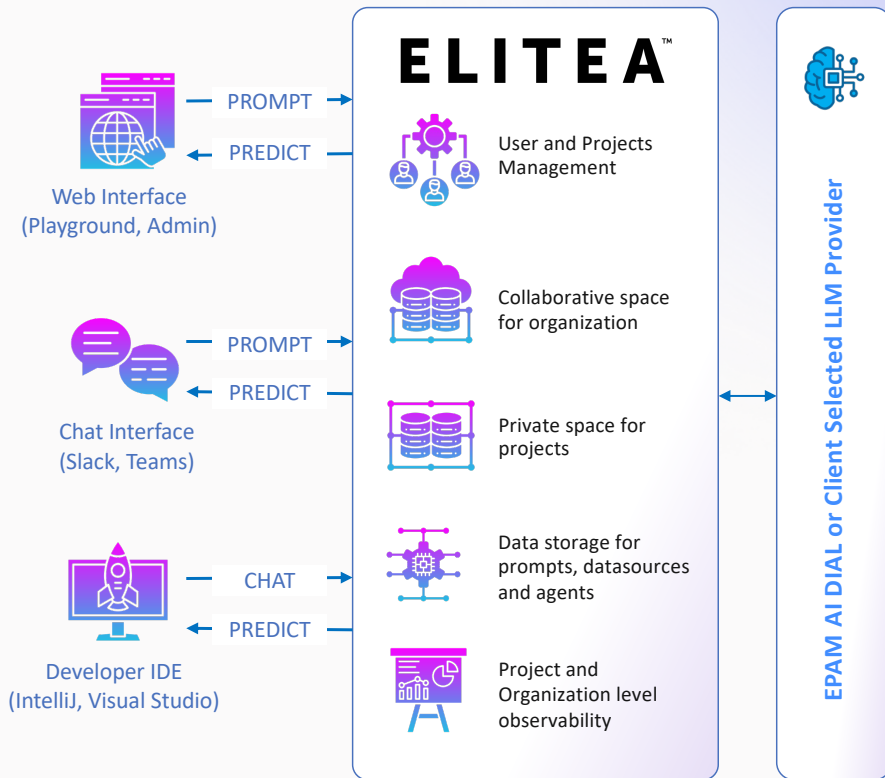


HOW DO YOU SUCCESSFULLY INTEGRATE GENERATIVE AI FOR  
SOFTWARE PRODUCTIVITY INTO THE LIFECYCLE?

# SECURE, COLLABORATIVE, ENTERPRISE-READY AI



## AI DIAL



# Prompt Engineering



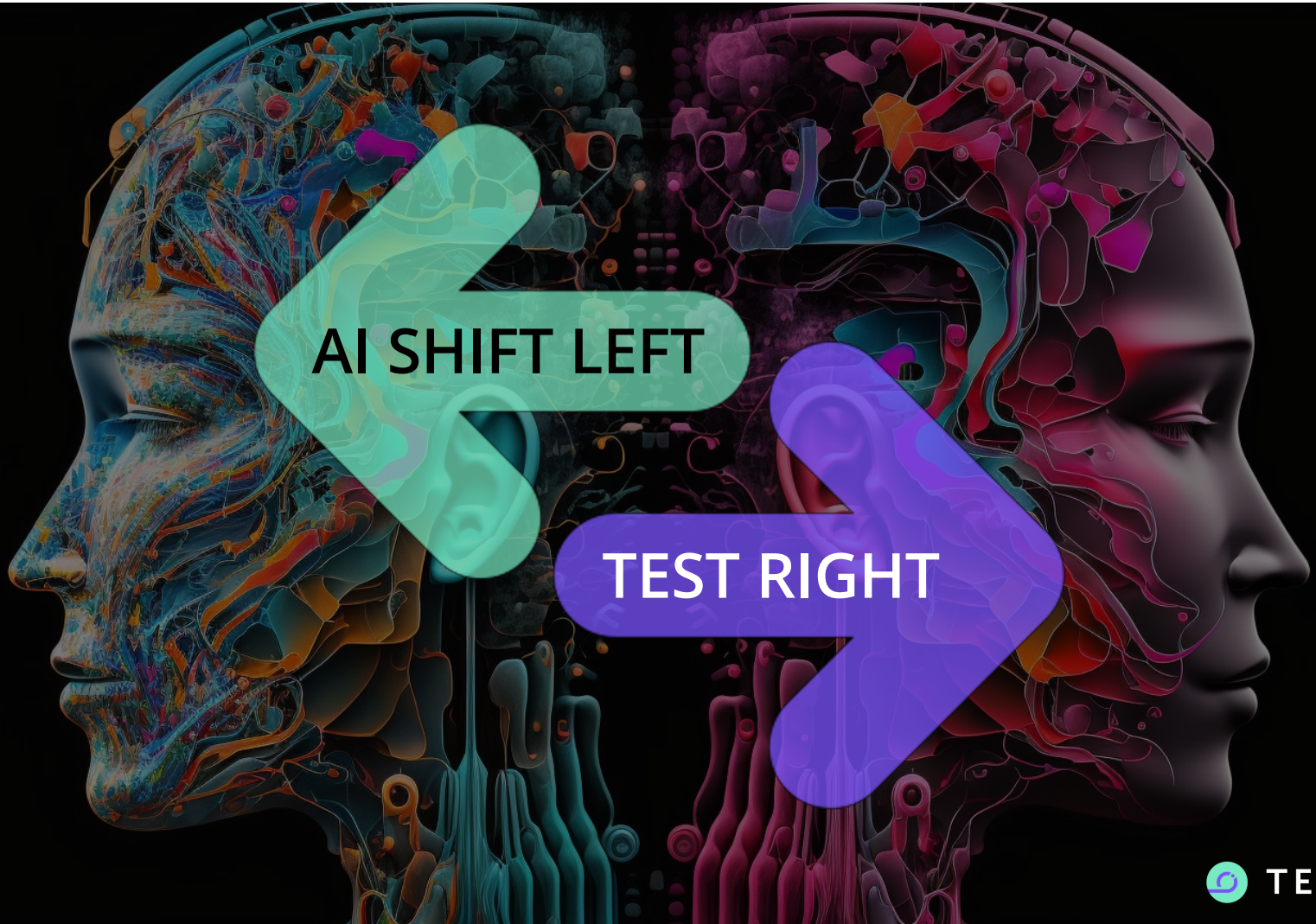


Artificial Intelligence  
**United**



brightest

Certified AI-Assisted **Test Engineer**

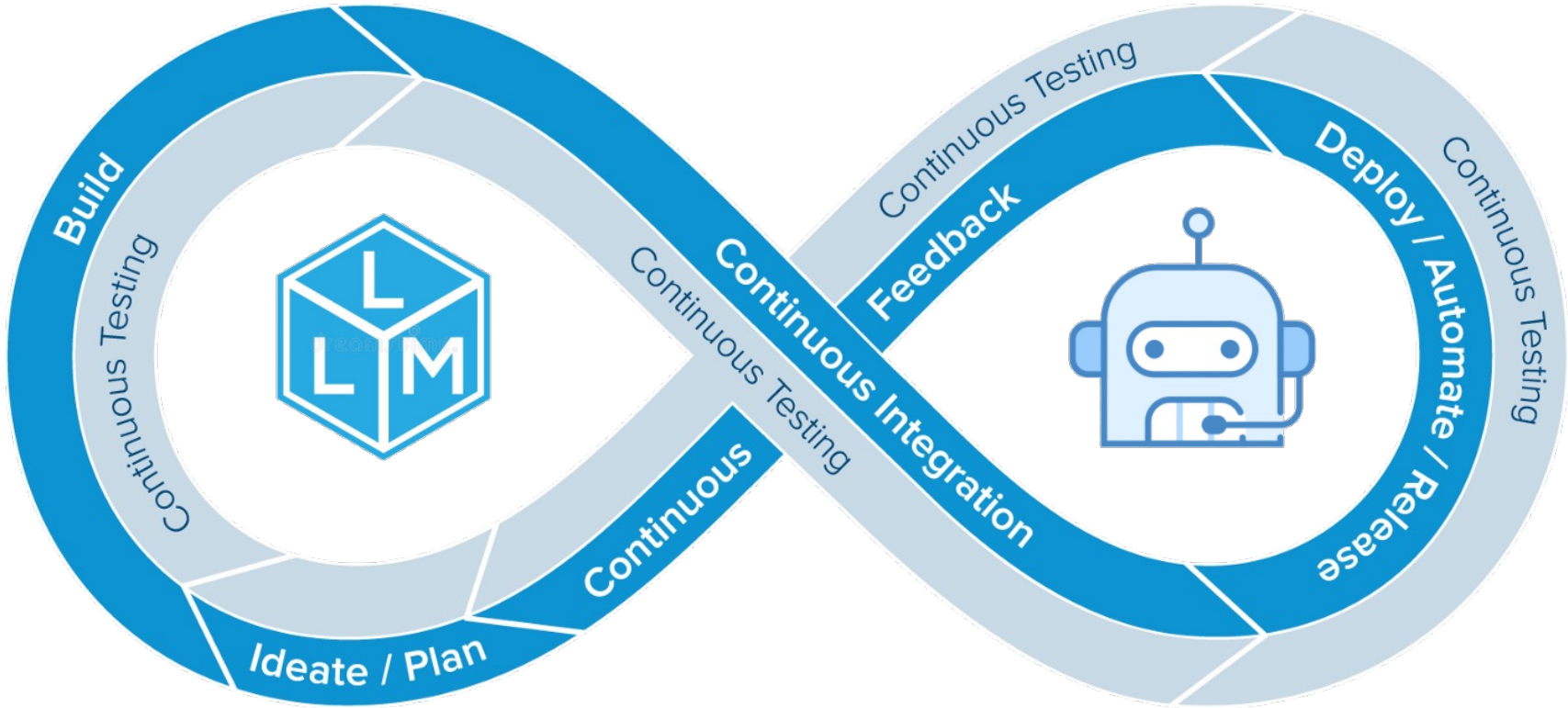


AI SHIFT LEFT

TEST RIGHT

# HOLISTIC QUALITY ENGINEERING FOR AI/ML





## EXAMPLE TESTS

Conduct user surveys, reviews, and checklists  
Hold usability sessions, e.g., task completion  
Measure the latency of model responses  
Test with screen readers, keyboard only, ...

User-Friendliness, User Satisfaction

Responsiveness, Accessibility

### Usability

Compare responses against ground truth  
Examine logical flow and structure of text  
Determine pertinence of output to input  
Check for contradictions in the text

Accuracy (Hallucinations)

Coherence, Relevance, Consistency

### Content Integrity

Analyze user interactions, e.g., time spent  
Assess uniqueness and originality of output  
Rate or calculate the readability of the output  
Test if response is appropriate for context

Interest Retention, Content Novelty

Fluency, Contextual Understanding

### Engagement

Check for equitable treatment of users/groups  
Attempt to generate offensive/harmful content  
Test if model reveals sensitive information  
Try to bypass authentication or authorization

Fairness, Safety, Privacy, Security

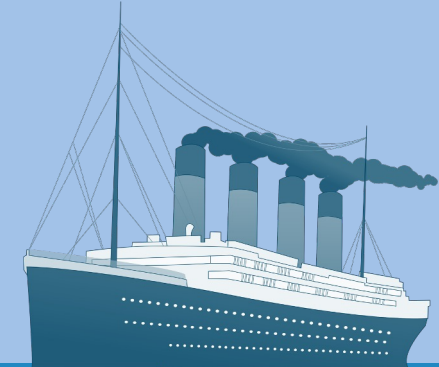
### Responsibility

Rate effectiveness of outputs in assisting users  
Assess range of responses for a given prompt  
Judge utility of information against benchmarks  
Test ability to summarize content efficiently

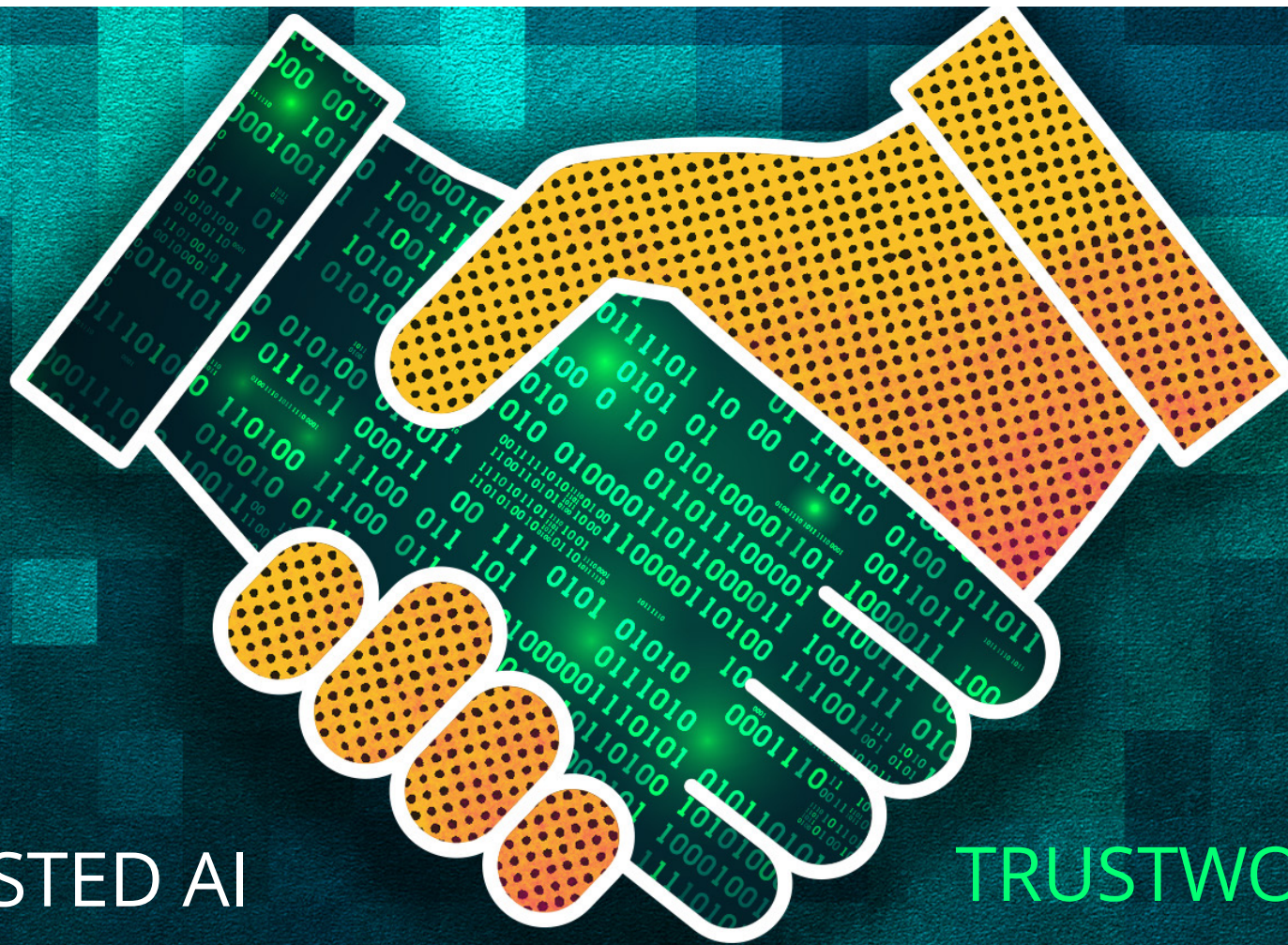
Usefulness, Diversity

Informativeness, Conciseness

### Practical Utility



BY TARIQ KING



TRUSTED AI

TRUSTWORTHY

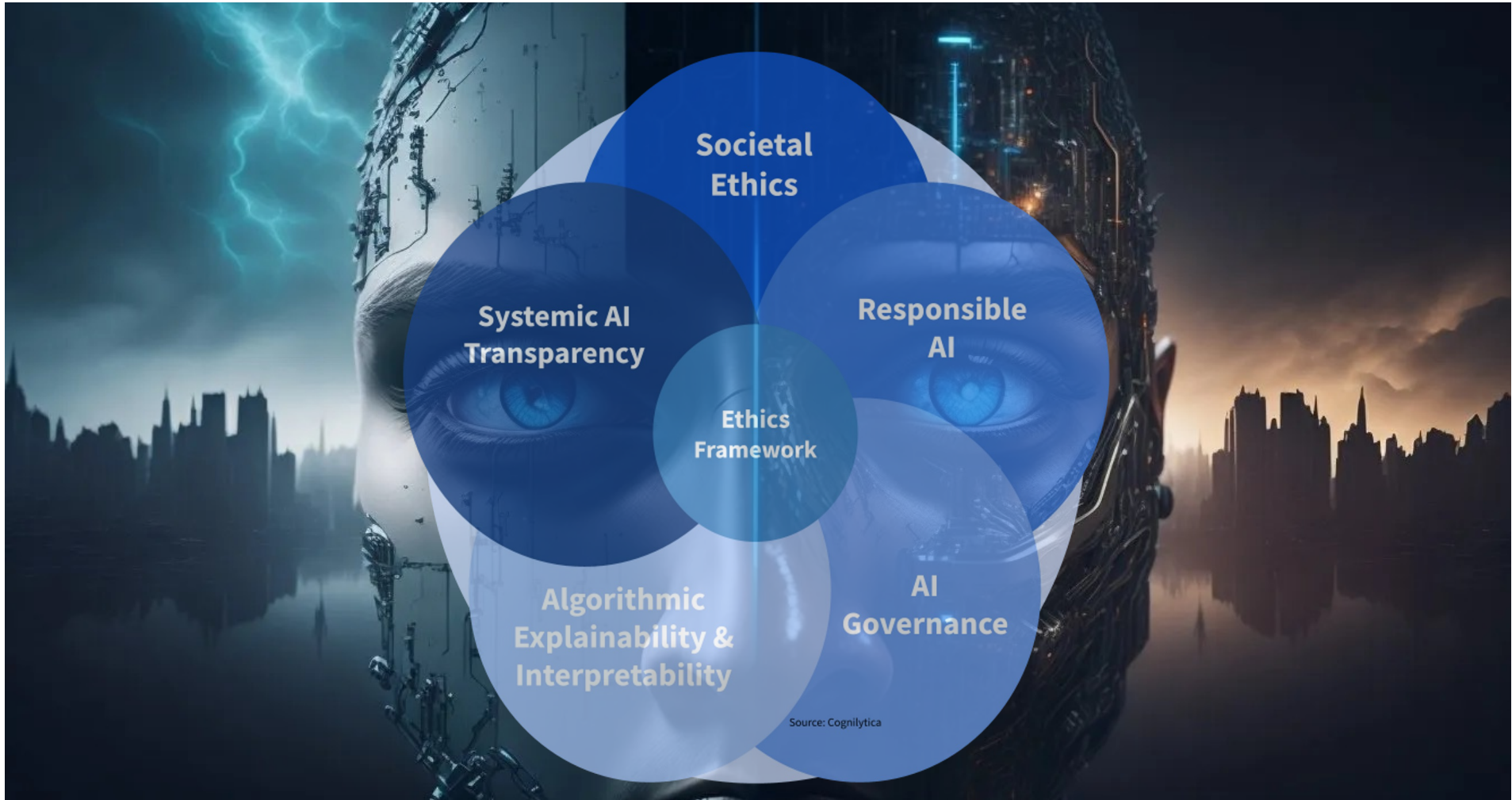
TESTABILITY

CONTROLLABILITY

OBSERVABILITY

EXPLAINABILITY





<epam>  TEST IO

# Thank You

For more information contact:

**Tariq King**

CEO and Head, Test IO

VP, Managing Principal, Technology, EPAM

[tariq\\_king@epam.com](mailto:tariq_king@epam.com)