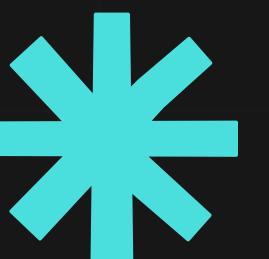


Hackathon
Presentation

Mainframe Code Explainability

Unlocking Legacy Systems for the Future



 ALGOWORKS



Presented
by Transformers++

What is Mainframe Code Explainability?

- The concept: Making legacy COBOL/Assembler code human-readable and understandable.
- Importance: Drives modernization, reduces dependency on retiring experts, and accelerates transformation.



Market Research & Trends

It's like having a superpower!



70% of global business transactions still run on mainframes.

Approximately 70% of global business transactions still rely on mainframes, underscoring their enduring role in critical enterprise operations.

\$65B market for mainframe modernization (Gartner 2024).

The mainframe modernization market is projected to reach \$65 billion in 2024, according to Gartner.

Key trends:

- Rise of AI-powered code explainability tools (e.g., IBM Watsonx Code Assistant, Open Legacy).
- Growing demand for cloud-native transformation.
- Increased regulatory focus on system transparency and auditability.





Business Challenges with Legacy Code

- Obsolete languages like COBOL with minimal documentation.
- Aging workforce and tribal knowledge at risk.
- High costs of maintenance and long onboarding times.
- Difficulty integrating legacy with modern APIs and platforms.
- Static systems vs agile market demands.



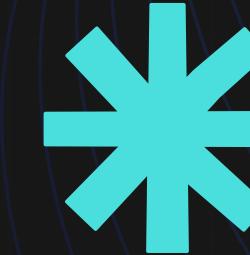


- **Client:** Leading BFSI enterprise on z/OS.
- **Problem:** Legacy COBOL code with minimal documentation and compliance risk.
- **Solution:**
 - Translate COBOL into readable pseudocode
 - Map business logic to UI workflows
 - Identify redundant modules for refactoring
- **Outcome:** 35% faster time to market and 60% better developer onboarding.

Use Case Scenario



How It Will Impact Algoworks



Business Value:

- Faster modernization of client systems
- Improve transformation transparency
- Increase service revenue

Technical Advantage:

- Reduce onboarding curve
- Reuse logic for DevOps and cloud

Strategic Edge:

- Expand into regulated industries



Features That Bring Clarity to Complexity



Key Features

Multi-Language: Python, C, C++, COBOL, Fortran, C#

Easy Upload & Preview: Drag-drop/browse (200 MB), live code view

Clear Explanations: Natural-language walkthroughs with logic highlights

Complexity Metrics: 1-10 score + visual chart

Vulnerability Flags: Exception, permission & crash-risk alerts

Auto Linting: formatting fixes, naming conventions, and readability improvements

Change Summary: “Key Changes” diff for quick refactoring

Export Reports: JSON for automation & polished PDF



Thank You For Your Attention

