Open Source With AI

Part 1

Facilitator: Alberto Camarena

Presented at: Global Hack Week 🌐

Date: 08/05/25 

🎯 Objectives

- Understand the landscape of Open Source AI
- Explore AI-driven **no-code/low-code** contributions
- Hands-on: AI-accelerated testing, documentation, and code comprehension
- Discuss best practices for AI contributions

1. Open Source × AI

What is Open Source AI?

- AI frameworks, models, datasets under open licenses
- Community model hubs (e.g., Hugging Face)
- But also accelerating open source projects with AI tools

AI that is Open Source

Category	Examples	Description
Frameworks	TensorFlow, PyTorch	Build AI models
Models	BERT, GPT-2	Pre-trained for diverse tasks
Datasets	ImageNet, COCO	Train and evaluate AI
Communities	Hugging Face Hub	Share models and datasets
Projects	OpenAI Gym, Stable Diffusion	Train and evaluate environments
Research	Papers with Code	Research papers and code
Standards	ONNX, PMML	Model interchange standards
Organizations	Linux Foundation AI	Promote open source AI

Why Open Source AI?

- Transparency & reproducibility
- Shared innovation & benchmarks

Contribution Modes

- Code & models
- Data curation & labeling
- Docs, examples & tutorials
- Evaluation & benchmarks

> *References:*
>
> Hugging Face: Contributing

2. Accelerating Open Source Projects with AI

AI Tools for Open Source

Tool	Description	Use Cases
GitHub Copilot	AI pair programmer	Code suggestions, refactoring
ChatGPT	Conversational AI	Code explanations, documentation
Diffblue	AI test generation	Unit tests for Java
Codeium	AI code completion	Multi-language code suggestions

In general, the great AI allies currently are LLMs (Large Language Models) that can help us with:

No-Code & Low-Code Contributions

Contribution Type	AI Tool	Usage Example
Documentation	Copilot / ChatGPT	“Write usage examples for function X”
Test Generation	ChatGPT / Diffblue	“Generate Jest tests for module Y”
Issue Triage	GitHub Actions + LLM	Auto-label new issues
Translations	DeepL / GPT-4	Translate README to Spanish
Code Refactoring	Copilot / ChatGPT	“Refactor this function for readability”
Code Comprehension	ChatGPT	“Explain this function”

3. Hands-On Exercises

Task 1: Generate Tests

1. Open provided JS module (no tests)
2. Use your favorite GenAI tool to create tests
3. Run and verify coverage

Task 2: Improve Documentation

1. Stub README.md
2. Prompt AI for installation & usage sections
3. Integrate into README
4. Do the same for code snippets

Task 3: Code Comprehension

1. Complex code analysis
2. Ask AI to explain and propose improvements
3. Validate suggestions manually

4. Best Practices & Cautions

- Always review AI output for correctness
- Anchor prompts in real code to avoid hallucinations
- Document AI usage in commit messages

Thank You!

Questions & Discussion.