

## Challenge, Team & Tiers

1 / 5

Team CodersBlue - All Tiers Complete

### Challenge Goal

"Bridge the semantic gap between human intent and system commands"

### Core Philosophy

Samantha listens to what you want in plain English, understands your intent, and safely gets things done.

"More than a tool—she's a helpful partner."

### The Problem We Solved

#### Steep Learning Curve

→ Natural language interface

#### Tedious Multi-step Tasks

→ Single command execution

#### Cognitive Overload

→ Focus on WHAT, not HOW

#### Team Members: CodersBlue

Khalid Elbagir  
Zayn Sharif  
Adam Azeb  
Abdulwahid Yaich  
Nabeel Ahmad

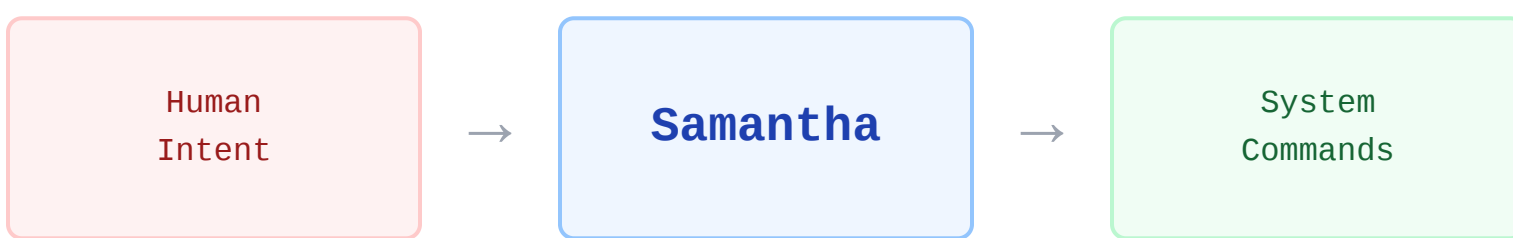
#### Tier Completion

- ✓ Tier 1: 100% Complete
- ✓ Tier 2: 100% Complete
- ✓ Tier 3: 100% Complete

## "Her"-Inspired Architecture

2 / 5

### Bridging the Semantic Gap



### Key Technical Innovations

- Modular Architecture (Parser, Planner, Executor)
- Dynamic JSON Plan Generation
- Context & Pronoun Resolution ("copy them")
- Full openEuler Docker compatibility
- Uses official Qwen3-Coder-480B & Qwen2.5-VL-32B

### Self-Correction Engine Example:

```
# User enters "buget"
correction = find_closest_match("buget", known_words)
# Samantha suggests 'budget'
=> "budget"
```

## Technical Architecture

3 / 5

### Core Components

CLI: src/cli/samantha.py

NL → Plan: core/nl2cmd.py

Executor: core/executor.py

Search: core/search.py

Memory: core/memory.py

Safety: core/safety.py

Suggestions: core/suggestions.py

openEuler: osint/openeuler.py

### Data Flow

- User enters a plain-English instruction.
- nl2cmd produces a structured JSON plan.
- safety validates paths & flags destructive steps for confirmation.
- executor runs steps with error recovery & self-correction.
- Result is returned with clear, conversational feedback.

### Runtime & Model Config

```
# Environment (as used in the README)
export OPENAI_API_KEY=EMPTY
export CODER_BASE_URL=http://10.10.171:8000/v1
export CODER_MODEL_NAME="Qwen/..."
```

## Results & Judging Criteria

4 / 5

### Judging Criteria Excellence

- ✓ **Functionality & Tier Completion (40%):** ALL 3 tiers implemented
- ✓ **Technical Sophistication (30%):** Advanced NLP + self-correction
- ✓ **UX & "Her" Inspiration (20%):** Intuitive conversational interface
- ✓ **openEuler Integration (10%):** Docker deployment ready

### Standout Achievements

- 🚀 **Professional Quality:** Full test suite with pytest.
- 🚀 **Safety-First Design:** Critical operations require user confirmation.
- 🚀 **Judge-Ready Demo:** A polished 2.5-minute presentation.
- 🚀 **Beyond Requirements:** All tiers + bonus features implemented.

### Live Demo - Challenge Fulfillment

```
# Tier 1: "Go to my documents folder"
→ Navigation ✓

# Tier 2: "Find PDFs I worked on yesterday"
→ Context ✓

# Tier 3: "Find files mentioning budget"
→ Content-aware ✓

# Bonus: "Organize my messy desktop"
→ Proactive intelligence ✓
```

## Live Demos in Action

5 / 5

### Demo 1: Self-Correction

```
(.venv) [root@...]:# python -m src.cli.samantha "search for buget in demo_data"
Samantha: I see you want to: 'search for buget...'.
Based on these assumptions:
- Searching for the term 'buget' (likely a typo for 'budget')...
I will perform the following steps:
1. search_in_files "budget" "demo_data"
Should I proceed with this plan? (y/n): y
--- Execution Summary ---
Step 1 [SUCCESS]: Found content:
/work/track1_CodersBlue/demo_data/q1-budget.txt:1:This file contains the quarterly budget.
...
```

### Demo 2: Multi-Step Operation

```
(.venv) [root@...]:# python -m src.cli.samantha "find pdf files in demo_data then move them to backup"
Samantha: I see you want to: 'find pdf files...'.
I will perform the following steps:
1. find_files "**.pdf" "demo_data"
2. mkdir "backup"
3. mv "{result_of_step_1}" "backup"
Should I proceed with this plan? (y/n): y
--- Execution Summary ---
Plan execution finished.
Step 1 [SUCCESS]
Step 2 [SUCCESS]
Step 3 [PARTIAL]
```

### Demo 3: Advanced Search (Bash Fallback)

```
(.venv) [root@...]:# python -m src.cli.samantha "find files larger than 1mb in demo_data"
Samantha: I see you want to: 'find files larger than 1mb...'.
I will perform the following steps:
1. execute_bash "find demo_data -type f -size +1M"
Reason: To search for all files larger than 1MB...
Should I proceed with this plan? (y/n): y
--- Execution Summary ---
Plan execution finished.
Step 1 [SUCCESS]
```