

# VCTT-AGI Phase 4.5 Demo Script (45 seconds)

---

**Target Duration:** 45 seconds

**Demo Date:** November 20, 2025

---

## Demo Flow (Timed)

---

### 0:00-0:10 — Introduction & Context (10s)

"This is MIN DeepAgent with the jazz team self-improvement loop.  
We just shipped 5 major features in a single sprint. Let me **show** you."

**Screen:** VCTT-AGI frontend homepage

---

### 0:10-0:20 — Feature 1: Backend Auto-Retry Guardrail (10s)

"First: The backend now automatically retries code edits if trust score drops below 75%.  
This ensures only high-quality autonomous code reaches production."

**Screen:**

- Show Render logs with retry attempts
  - Point to "Retry attempt 2/3 (previous  $\tau < 0.75$ )" log line
- 

### 0:20-0:30 — Feature 2: Jazz Team Analysis UI (10s)

"Second: Real-time jazz team analysis with Voice, Choice, Transparency, Trust scores,  
plus an 'Apply Refined' button that auto-improves your instruction."

**Screen:**

- Open DeepAgent IDE
  - Trigger Cmd+K on sample code
  - Show jazz analysis panel with colored metrics
  - Click "Apply Refined" button → show it auto-submits
-

## 0:30-0:40 — Features 3-5: Enhanced Prompts, Drag & Drop, Search (10s)

"Third: Enhanced prompts with 5 jazz-learned best practices.  
Fourth: Drag-and-drop file tree.  
Fifth: Cmd+Shift+F search across all files."

### Screen (rapid cuts):

- Show code edit response with better documentation (prompt enhancement)
- Drag a file from one folder to another
- Open search modal, type "function", show results

## 0:40-0:45 — Closing & Call to Action (5s)

"All features live on Render + Vercel. Ready to ship public beta.  
Deploy now?"

### Screen:

- Show both deployment URLs:
- Backend: <https://vctt-agi-backend.onrender.com>
- Frontend: <https://vcttagiui.vercel.app> (or your Vercel URL)



## Pre-Demo Checklist

- ☐ Both Render and Vercel deployments are live and healthy
- ☐ Sample code file ready in editor for Cmd+K demo
- ☐ Browser tabs prepped: Frontend, Render logs, Vercel dashboard
- ☐ Screen recording software ready (Loom, OBS, etc.)
- ☐ Test jazz analysis endpoint: `/api/ide/code-edit` returns full `jazzAnalysis` object
- ☐ Clear browser cache to ensure latest frontend assets load



## Demo Tips

1. **Speak clearly and pace yourself** - 45 seconds goes fast
2. **Show, don't tell** - Visual proof > verbal claims
3. **Keep cursor movements smooth** - No frantic clicking
4. **Use keyboard shortcuts** - Looks professional (Cmd+K, Cmd+Shift+F)
5. **End with energy** - "Ready to ship!" creates momentum



## Post-Demo Next Steps

1. **User Reply:** "Ship it"

2. **Action:** Deploy to public beta URLs
  3. **Announce:** Share demo + links in relevant channels
  4. **Monitor:** Watch production logs for first 24h
- 



## What We Shipped (Sprint Summary)

---



### Completed (5/12 tasks)

1. **Backend Guardrail:** Auto-retry if  $\tau < 0.75$  (3 retries, exponential backoff)
2. **Frontend Jazz UI:** Voice/Choice/Transparency/Trust scores + “Apply Refined” button
3. **Prompt Optimizations:** Merged jazz team top 5 best practices into code edit prompts
4. **Drag & Drop:** Full file/folder drag & drop in file tree
5. **Search in Files:** Cmd+Shift+F modal with case-sensitive + regex options



### Deferred (MVP versions for future sprint)

1. Minimap + folding
2. Git gutter
3. Command palette (Cmd+P)
4. Test explorer
5. Deployment panel



### Meta Tasks

1. **Zero-Error Sweep:** All code builds cleanly
  2. **Loom Demo:** This script
- 



## Key Selling Points

---

1. **Self-Improving AI:** Jazz team analyzes its own outputs and suggests improvements
  2. **Production-Grade Reliability:** Auto-retry guardrail ensures high trust scores
  3. **Transparency:** Real-time metrics (Voice/Choice/Transparency/Trust) build user confidence
  4. **Best Practices Baked In:** 5 jazz-learned patterns automatically applied to every code edit
  5. **Full-Stack Solution:** React frontend + NestJS backend, deployed on Render + Vercel
- 

Ready to record! 🎬