VOICE PRD TRUE METHOD SYSTEM

10-Minute Voice → Manual Iterative Execution with Fresh Context

Purpose: Transform Wispr Flow dictation into atomic user stories executed with ZERO context rot through fresh Claude Code sessions.

THE PHILOSOPHY

The TRUE method prevents context rot through fresh sessions.

An autonomous agent executing in a degraded context will:

- ❌ Make compounding errors as context degrades
- ❌ Lose track of requirements halfway through
- ❌ Produce inconsistent code across iterations
- ❌ Burn tokens on garbage output

The TRUE method:

- ✅ Each user story = ONE fresh Claude Code session
- ✅ Maximum context window = optimal performance
- ✅ Manual validation between stories catches issues early
- ✅ No plugin automation = no hidden context rot

WISPR FLOW DICTATION SCRIPT

Before You Press Record:

Have Ready:

- Project folder path (absolute path)
- Any API keys/credentials you know about
- Links to any docs/examples you want referenced
- Existing codebase context if adding to existing project

Recording Structure (~10-12 minutes):

BLOCK 1: PROJECT CORE (2 min)

"The project is called [NAME]. This is for [business]. The whole point of this is [explain WHY]. Success means [concrete outcome]. Timeline is [urgent/normal/exploratory]."

BLOCK 2: THE PROBLEM (2 min)

"Right now, [describe pain state]. What's broken is [specific failure]. The manual process involves [steps]. Without this, [consequence]. The trigger is [recent event]."

BLOCK 3: THE SOLUTION (3 min)

"What I want to build is [high level]. Core features: Feature 1, Feature 2, Feature 3. Main workflow is [happy path]. Tech stack: [be explicit]. UI: [Tailwind/Shadcn/etc]. Database: [Postgres/Supabase/etc]. Auth: [method]. File structure: [organization]."

BLOCK 4: INTEGRATION POINTS (1 min)

"Integrates with [API/service]. Credentials are [location]. API docs: [URL]. Rate limits: [if known]. Error handling: [fail gracefully/retry/alert]."

BLOCK 5: EDGE CASES & CONSTRAINTS (2 min)

"What happens when [unexpected thing]? Handle [error scenarios]. Should NOT [anti-features]. Performance needs: [requirements]. Mobile: [responsive/desktop only]. Critical constraint: [budget/time/limits]."

BLOCK 6: SUCCESS CRITERIA (1 min)

"Done when: [Concrete tests]. Quality bar: [MVP/production/agency]. Testing: [manual/unit/integration]. Documentation: [README/full docs/tutorial]."

CLAUDE'S CLARIFICATION PROTOCOL

STEP 1: INITIAL TRIAGE

Parse transcript into:

- **DECLARED:** Explicit technical choices, APIs, constraints
- **IMPLIED:** Standard patterns, defaults, best practices
- **MISSING_BLOCKERS:** File paths, auth, credentials, schemas
- **MISSING_DETAILS:** Error handling, UI/UX, testing, deployment
- **AMBIGUOUS:** Contradictions, vague scope, unclear dependencies
- **RISKS:** Complex integrations, performance requirements

STEP 2: SMART DEFAULTS

Declare intelligent defaults before asking questions.

STEP 3: DEPENDENCY-ORDERED QUESTIONS

Ask ONE question at a time. Wait for answer. Then next.

Priority Tiers:

- 🔴 **TIER 0:** Absolute blockers (paths, auth, credentials)
- 🟡 **TIER 1:** Implementation details (error handling, data flow)
- 🟢 **TIER 2:** Polish & preferences (propose, don't ask)

Target: 5-10 questions maximum

THE TRUE METHOD EXECUTION

PRINCIPLES:

1. Each user story = ONE fresh Claude Code session
2. Manual validation after each story
3. NO plugin automation
4. Stories must be atomic
5. Sequential execution only

EXECUTION WORKFLOW:

For Each User Story:

1. **Open Fresh Claude Code Session**
2. **Provide Context** (paste story + project context)
3. **Let Claude Execute** (don't interrupt)
4. **Validate Results** (test acceptance criteria)
5. **Record Handoff** (timestamp, files, gotchas)
6. **Move to Next Story** (close session, open fresh)

CRITICAL RULES:

- ✅ ALWAYS start fresh session per story
- ✅ ALWAYS validate before next story
- ✅ NEVER reuse degraded context
- ✅ NEVER multiple stories in one session
- ❌ NEVER use plugins or automation
- ❌ NEVER assume completion without testing

SAFETY LIMITS

- **Max Stories Per Session:** 1
- **Session Timeout:** 10 minutes
- **Total Budget:** $30 estimated
- **Validation:** After EVERY story

ERROR RECOVERY

If Story Fails:

1. Stop immediately
2. Analyze failure mode
3. DO NOT continue in same session
4. Fix root cause
5. Start FRESH session and retry

If Validation Fails:

1. DO NOT move to next story
2. Open FRESH session
3. Retry with adjusted criteria
4. Validate again

5. Only proceed when passing

SUCCESS METRICS

TRUE Method Excellence:

✅ Each story completes in ONE fresh session ✅ Zero context degradation ✅ Validation passes after each story ✅ No backtracking or rework ✅ Final product meets all criteria ✅ Execution time < estimated ✅ Budget within limits

---

WHY THIS WORKS

1. Fresh context = fresh thinking

- No accumulated errors
- No forgotten context
- Optimal token usage

2. Manual validation catches issues early

- Before problems compound
- When cheap to fix
- With full understanding

3. Atomic stories = clear scope

- One goal per session
- Complete or not (no partial)
- Easy to parallelize later

4. No hidden automation = no hidden failures

- You see every step
- You validate every result
- You control the process

This method trades automation speed for execution quality.

For autonomous agents, this is the gold standard for reliability.