PHASE 2: E&P ASSESSMENT MULTI-AGENT INTEGRATION

Epic-Level Project Plan

PROJECT OVERVIEW

Project Name: E&P Assessment Suggestibility Adapter Agent
Phase: 2 of 4
Platform Integration: HypnoSelfImprovementPlatform + HypnoResilientMind
Ultimate Vision: Integrated Holistic Mental Wellness XR/VR Hub

EPIC BREAKDOWN

EPIC 1: Suggestibility Adapter Agent Foundation

Goal: Create intelligent agent that adapts communication based on E&P Assessment results
Dependencies: Phase 1 E&P Assessment API (COMPLETE)
Timeline: 2-3 weeks
Priority: CRITICAL

EPIC 2: Multi-Agent Orchestration Integration

Goal: Integrate adapter agent with AutoGen/CrewAI multi-agent systems
Dependencies: Epic 1
Timeline: 2 weeks
Priority: HIGH

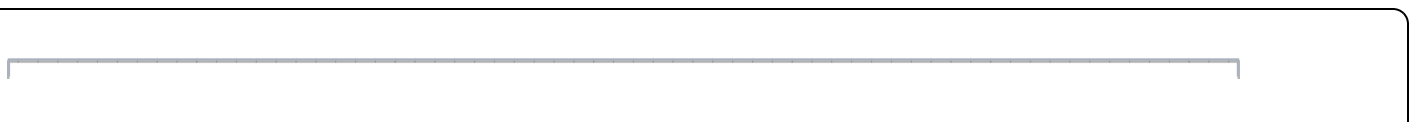
EPIC 3: Real-Time Language Pattern Adaptation

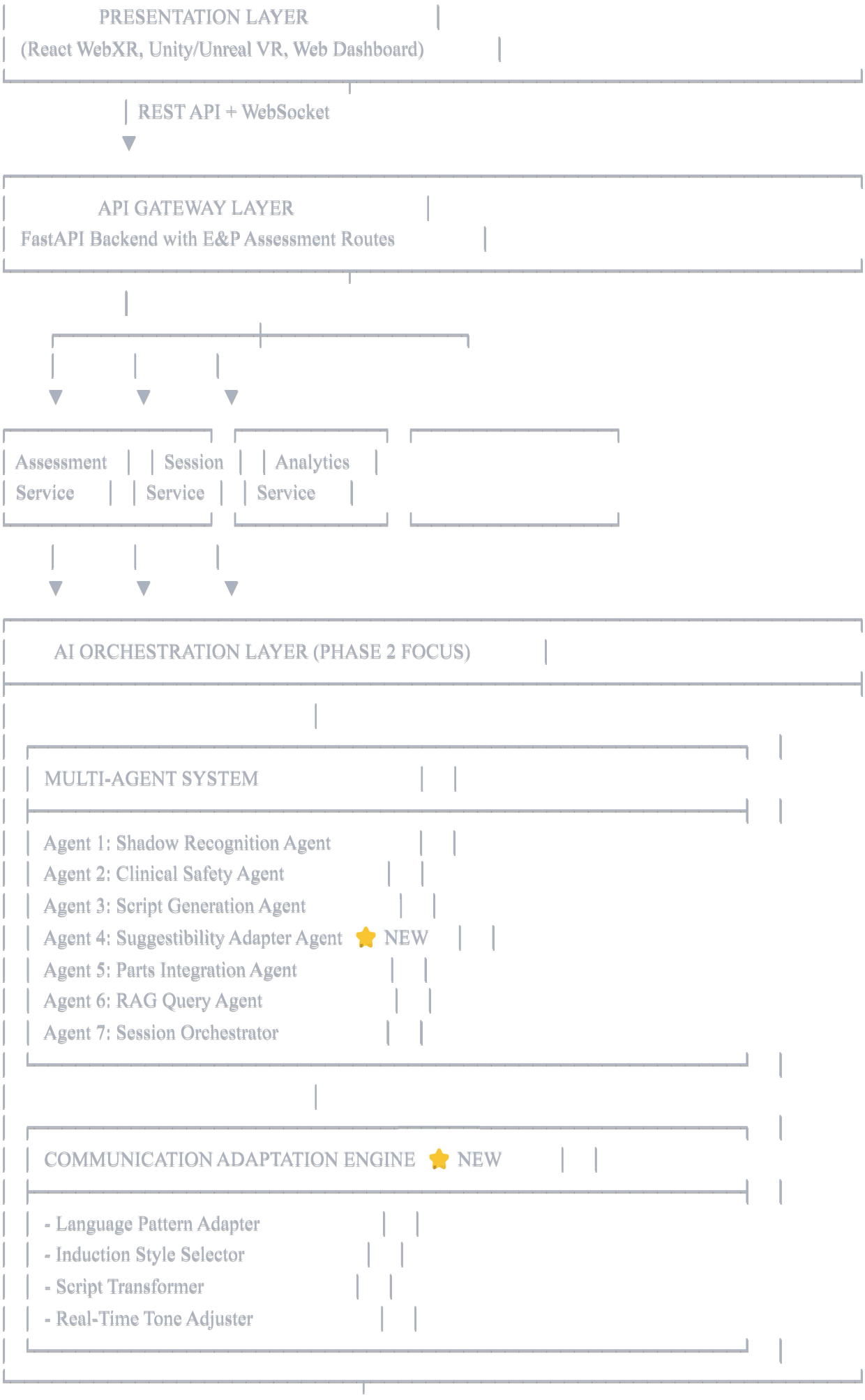
Goal: Dynamic script modification based on suggestibility type
Dependencies: Epic 1, Epic 2
Timeline: 1-2 weeks
Priority: HIGH

EPIC 4: Quality Assurance & Clinical Safety

Goal: Ensure safe, effective, and clinically appropriate adaptation
Dependencies: Epic 1-3
Timeline: 1 week
Priority: CRITICAL

ARCHITECTURE LAYERS




PRESENTATION LAYER
(React WebXR, Unity/Unreal VR, Web Dashboard)
REST API + WebSocket
API GATEWAY LAYER
FastAPI Backend with E&P Assessment Routes
Assessment Service
Session Service
Analytics Service
AI ORCHESTRATION LAYER (PHASE 2 FOCUS)
MULTI-AGENT SYSTEM
Agent 1: Shadow Recognition Agent
Agent 2: Clinical Safety Agent
Agent 3: Script Generation Agent
Agent 4: Suggestibility Adapter Agent ⭐ NEW
Agent 5: Parts Integration Agent
Agent 6: RAG Query Agent
Agent 7: Session Orchestrator
COMMUNICATION ADAPTATION ENGINE ⭐ NEW
- Language Pattern Adapter
- Induction Style Selector
- Script Transformer
- Real-Time Tone Adjuster

EPIC 1: SUGGESTIBILITY ADAPTER AGENT FOUNDATION

Feature 1.1: E&P Preferences Client

Description: Service client to fetch user's communication preferences from E&P API

Story 1.1.1: Create API Client

As a Multi-Agent System
I want to fetch user E&P preferences
So that agents can adapt communication style

Acceptance Criteria:

- Client can call GET /api/v1/assessment/ep/communication-preferences
- Handles authentication (JWT)
- Caches preferences (Redis) for 1 hour
- Handles missing assessments gracefully

Tasks:

- [ ] Task 1.1.1.1: Create EPPreferencesClient class
- [ ] Task 1.1.1.2: Implement HTTP client with retry logic
- [ ] Task 1.1.1.3: Add Redis caching layer
- [ ] Task 1.1.1.4: Write unit tests
- [ ] Task 1.1.1.5: Create integration tests

Estimated: 1 day

Story 1.1.2: Preferences Response Parser

As a Adapter Agent
I want to parse E&P preferences into structured data
So that I can make adaptation decisions

Acceptance Criteria:

- Parses JSON response from E&P API

- Extracts suggestibility_type, language_patterns, preferred_inductions
- Validates data completeness
- Returns typed Python dataclass

Tasks:

- [ ] Task 1.1.2.1: Define PreferencesData dataclass
- [ ] Task 1.1.2.2: Implement JSON parser
- [ ] Task 1.1.2.3: Add validation logic
- [ ] Task 1.1.2.4: Write parser tests

Estimated: 0.5 days

Feature 1.2: Language Pattern Adapter

Description: Core engine that transforms generic text to suggestibility-specific language

Story 1.2.1: Physical Suggestible Adapter

As a Script Generator
I want to convert text to Physical-appropriate language
So that Physical suggestibles receive literal, direct communication

Acceptance Criteria:

- Converts inferential to direct language
- Removes metaphors, replaces with literal descriptions
- Emphasizes body sensations
- Uses present-tense, authoritative tone

Example:

```
Input:  "You might find yourself relaxing..."
Output: "Close your eyes. Relax now."

Input:  "Like a feather floating down..."
Output: "Feel your body becoming heavy. Notice the weight."
```

Tasks:

- [ ] Task 1.2.1.1: Define Physical transformation rules
- [ ] Task 1.2.1.2: Implement pattern matching & replacement
- [ ] Task 1.2.1.3: Create transformation pipeline

- [ ] Task 1.2.1.4: Write transformation tests (20+ examples)
- [ ] Task 1.2.1.5: Benchmark performance

Estimated: 2 days

Story 1.2.2: Emotional Suggestible Adapter

As a Script Generator
I want to convert text to Emotional-appropriate language
So that Emotional suggestibles receive indirect, metaphorical communication

Acceptance Criteria:

- Converts direct to inferential language
- Adds metaphorical imagery
- Emphasizes emotions and feelings
- Uses past/future tense, permissive tone

Example:

```
Input:  "Close your eyes. Relax now."
Output: "You might find your eyelids becoming heavy... almost as if they prefer to close..."

Input:  "Feel your body becoming heavy."
Output: "Perhaps you'll discover a sense of gentle heaviness... like a warm blanket..."
```

Tasks:

- [ ] Task 1.2.2.1: Define Emotional transformation rules
- [ ] Task 1.2.2.2: Implement metaphor generation
- [ ] Task 1.2.2.3: Create transformation pipeline
- [ ] Task 1.2.2.4: Write transformation tests (20+ examples)
- [ ] Task 1.2.2.5: Benchmark performance

Estimated: 2 days

Story 1.2.3: Somnambulistic (Balanced) Adapter

As a Script Generator
I want to use balanced approach
So that Somnambulistic individuals receive appropriate mixed communication

Acceptance Criteria:

- Mixes direct and inferential language

- Balances literal and metaphorical
- Alternates between body and emotion focus
- Uses flexible tense and tone

Tasks:

- [ ] Task 1.2.3.1: Define balanced rules
- [ ] Task 1.2.3.2: Implement mixed-approach transformer
- [ ] Task 1.2.3.3: Write tests

Estimated: 1 day

Feature 1.3: Induction Style Selector

Description: Selects appropriate hypnotic induction based on suggestibility type

Story 1.3.1: Induction Template Library

As a Adapter Agent
I want to have pre-defined induction templates for each type
So that I can select clinically-appropriate techniques

Acceptance Criteria:

- Template library with 3+ inductions per type
- Physical: Progressive Relaxation, Eye Fixation, Body Scan
- Emotional: Guided Imagery, Metaphorical Journey, Confusion
- Somnambulistic: Dave Elman, Rapid, Combined

Tasks:

- [ ] Task 1.3.1.1: Create InductionTemplate dataclass
- [ ] Task 1.3.1.2: Write 9+ induction templates (HMI-based)
- [ ] Task 1.3.1.3: Validate with clinical hypnotherapist
- [ ] Task 1.3.1.4: Store in database
- [ ] Task 1.3.1.5: Create retrieval service

Estimated: 2 days

Story 1.3.2: Dynamic Induction Selector

As a Adapter Agent
I want to select best induction based on user profile
So that each session uses optimal technique

Acceptance Criteria:

- Considers: suggestibility type, confidence score, past responses
- Returns ranked list of suitable inductions
- Logs selection reasoning

Tasks:

- [ ] Task 1.3.2.1: Implement selection algorithm
- [ ] Task 1.3.2.2: Add confidence-based filtering
- [ ] Task 1.3.2.3: Create ranking system
- [ ] Task 1.3.2.4: Write selector tests

Estimated: 1 day

Feature 1.4: Adapter Agent Core
Description: Main agent class that orchestrates E&P-based adaptation

Story 1.4.1: Create SuggestibilityAdapterAgent Class

As a Multi-Agent Orchestrator
I want to instantiate adapter agent
So that it can participate in multi-agent workflows

Acceptance Criteria:

- Compatible with AutoGen AssistantAgent interface
- Compatible with CrewAI Agent interface
- Has tools: adapt_script, select_induction, get_preferences
- Maintains conversation context
- Logs all adaptations

Tasks:

- [ ] Task 1.4.1.1: Define agent interface
- [ ] Task 1.4.1.2: Implement AutoGen version
- [ ] Task 1.4.1.3: Implement CrewAI version
- [ ] Task 1.4.1.4: Create tool wrappers
- [ ] Task 1.4.1.5: Add logging/monitoring
- [ ] Task 1.4.1.6: Write integration tests

Estimated: 3 days

EPIC 2: MULTI-AGENT ORCHESTRATION INTEGRATION

Feature 2.1: AutoGen Integration (ResilientMind Platform)

Description: Integrate adapter agent with existing AutoGen multi-agent system

Story 2.1.1: Add Adapter to Hypno Team

As a ResilientMind Orchestrator
I want to include Suggestibility Adapter in agent team
So that all scripts are personalized

Acceptance Criteria:

- Adapter agent joins AutoGen RoundRobinGroupChat
- Receives user_id at session start
- Fetches E&P preferences before script generation
- Passes preferences to Script Generation Agent

Tasks:

- [ ] Task 2.1.1.1: Modify build_hypno_team() to include adapter
- [ ] Task 2.1.1.2: Update team workflow (adapter runs first)
- [ ] Task 2.1.1.3: Pass preferences via message context
- [ ] Task 2.1.1.4: Test full workflow
- [ ] Task 2.1.1.5: Verify tool calls in logs

Estimated: 2 days

Story 2.1.2: RAG Integration with E&P

As a RAG Query Agent
I want to filter results by suggestibility type
So that retrieved content is appropriate

Acceptance Criteria:

- RAG queries tagged with suggestibility_type
- Database stores content suitability metadata
- Search results filtered/ranked by appropriateness

Tasks:

- [ ] Task 2.1.2.1: Add suggestibility_tags to hypno_documents table
- [ ] Task 2.1.2.2: Update RAG query to accept filter parameter
- [ ] Task 2.1.2.3: Implement filtered search

- [ ] Task 2.1.2.4: Test with different types

Estimated: 1.5 days

Feature 2.2: CrewAI Integration

Description: Integrate adapter agent with CrewAI orchestration

Story 2.2.1: Create CrewAI Adapter Agent

As a CrewAI Orchestrator
I want to use adapter as CrewAI agent
So that I can leverage Crew features

Acceptance Criteria:

- Adapter implements CrewAI Agent interface
- Has defined role, goal, backstory
- Tools work with CrewAI task system
- Integrates with other crew members

Tasks:

- [ ] Task 2.2.1.1: Create CrewAI wrapper class
- [ ] Task 2.2.1.2: Define agent role/goal/backstory
- [ ] Task 2.2.1.3: Implement CrewAI tools
- [ ] Task 2.2.1.4: Write integration tests

Estimated: 2 days

Story 2.2.2: Session Generation Crew with Adapter

As a Therapeutic Session Orchestrator
I want to multi-agent crew with adapter included
So that sessions are personalized and clinically sound

Acceptance Criteria:

- Crew includes: Safety Agent, Adapter Agent, Script Generator
- Sequential workflow: Safety → Adapter → Generator
- Adapter findings passed to downstream agents

Tasks:

- [ ] Task 2.2.2.1: Define crew structure
- [ ] Task 2.2.2.2: Create sequential workflow

- [ ] Task 2.2.2.3: Implement task handoffs
- [ ] Task 2.2.2.4: Test full crew execution

Estimated: 2 days

Feature 2.3: Session Orchestrator Enhancement

Description: Update session orchestrator to use adapter

Story 2.3.1: Pre-Session E&P Check

As a Session Orchestrator
I want to verify E&P assessment completion before session
So that personalization is always available

Acceptance Criteria:

- Checks if user has completed E&P assessment
- Prompts completion if missing
- Loads preferences before agent initialization
- Caches preferences for session duration

Tasks:

- [ ] Task 2.3.1.1: Add E&P check to session startup
- [ ] Task 2.3.1.2: Implement completion prompt UI
- [ ] Task 2.3.1.3: Cache preferences in session state
- [ ] Task 2.3.1.4: Add fallback for missing assessment

Estimated: 1.5 days

EPIC 3: REAL-TIME LANGUAGE PATTERN ADAPTATION

Feature 3.1: Script Transformation Pipeline

Description: End-to-end pipeline for transforming generic scripts to personalized

Story 3.1.1: Multi-Stage Transformer

As a Script Generator
I want to transform script through multiple stages
So that adaptation is comprehensive and clinically appropriate

Stages:

1. Parse script structure (intro, induction, deepening, suggestions, emergence)

2. Apply language pattern transformation
3. Adjust pacing and pauses
4. Verify clinical safety
5. Format output

Tasks:

- [ ] Task 3.1.1.1: Define pipeline stages
- [ ] Task 3.1.1.2: Implement stage interfaces
- [ ] Task 3.1.1.3: Create pipeline orchestrator
- [ ] Task 3.1.1.4: Add stage validation
- [ ] Task 3.1.1.5: Write pipeline tests

Estimated: 2 days

Feature 3.2: Confidence-Based Adaptation

Description: Adjust adaptation strength based on E&P confidence score

Story 3.2.1: Adaptive Transformation Strength

As a Adapter Agent
I want to use lighter adaptation for low-confidence assessments
So that uncertain data doesn't over-influence output

Rules:

- Confidence >= 80%: Full transformation
- Confidence 60-79%: Moderate transformation
- Confidence < 60%: Minimal transformation + flag for review

Tasks:

- [ ] Task 3.2.1.1: Implement confidence-based scaling
- [ ] Task 3.2.1.2: Define transformation strength levels
- [ ] Task 3.2.1.3: Add clinical review flagging
- [ ] Task 3.2.1.4: Test with various confidence scores

Estimated: 1 day

EPIC 4: QUALITY ASSURANCE & CLINICAL SAFETY

Feature 4.1: Clinical Safety Validation

Description: Ensure all adapted content is clinically appropriate

Story 4.1.1: Contraindication Checker

As a Clinical Safety Agent
I want to validate adapted scripts against contraindications
So that harmful content is never produced

Acceptance Criteria:

- Checks for: re-traumatization risks, inappropriate suggestions, unsafe pacing
- Validates induction appropriateness
- Flags for clinician review if needed

Tasks:

- [ ] Task 4.1.1.1: Define contraindication rules
- [ ] Task 4.1.1.2: Implement checker logic
- [ ] Task 4.1.1.3: Integrate with adapter workflow
- [ ] Task 4.1.1.4: Create safety tests

Estimated: 2 days

Feature 4.2: Adaptation Quality Metrics

Description: Track and monitor adaptation effectiveness

Story 4.2.1: Metrics Collection

As a Platform Administrator
I want to track adaptation quality metrics
So that I can improve system performance

Metrics:

- Adaptation rate (% sessions using adapter)
- Confidence distribution
- User satisfaction by suggestibility type
- Clinical review rate

Tasks:

- [ ] Task 4.2.1.1: Define metrics schema

- [ ] Task 4.2.1.2: Implement collection points
- [ ] Task 4.2.1.3: Create analytics dashboard
- [ ] Task 4.2.1.4: Set up alerting

Estimated: 2 days

TIMELINE SUMMARY

Epic	Duration	Dependencies
Epic 1: Foundation	2-3 weeks	Phase 1 complete
Epic 2: Integration	2 weeks	Epic 1
Epic 3: Adaptation	1-2 weeks	Epic 1, 2
Epic 4: Safety/QA	1 week	Epic 1-3
TOTAL	**6-8 weeks**	Sequential

INTEGRATION WITH RESILIENTMIND PLATFORM

Current ResilientMind Architecture:

```
User Question → FastAPI Core → AutoGen Agent → MCP RAG Server → PostgreSQL
```

With Phase 2 Integration:

```
User Question → FastAPI Core →
  ↓
  1. E&P Preferences Client → E&P Assessment API
  ↓
  2. AutoGen Team (with Adapter Agent) →
     - Safety Agent (checks contraindications)
     - Adapter Agent (fetches E&P, adapts approach) ⭐ NEW
     - RAG Agent (retrieves filtered content)
     - Script Generator (produces personalized script)
  ↓
  3. MCP RAG Server → PostgreSQL (filtered by suggestibility)
  ↓
  4. Response (personalized, adapted, safe)
```

SUCCESS CRITERIA

Phase 2 is successful when:

- [ ] Adapter agent operational in both AutoGen and CrewAI
- [ ] 90%+ of sessions use E&P-based personalization
- [ ] Language patterns appropriately adapted (validated by clinicians)
- [ ] No adverse clinical safety events
- [ ] User satisfaction increases by 25%+
- [ ] Integration tests pass for both platforms

NEXT STEPS

Immediate Actions:

1. Review this plan with clinical team
2. Set up project management (Jira/Linear)
3. Create development branch
4. Begin Epic 1, Feature 1.1 (E&P Preferences Client)

Ready to start Phase 2 implementation?