

# New Workflow Requirements Analysis

---

## Comparison: Current Flow vs. New Requirements

---

**Date:** December 2025

**Document Purpose:** Analyze differences between existing workflow and new requirements, identify implementation requirements, and assess complexities.






---

## Executive Summary

---

You have requested a new workflow that shifts from **admin-mediated validation** to **user-driven validation** with an intermediate **synopsis generation step** before final output. The new workflow introduces **shot list generation** instead of script generation, and includes an **iteration loop** for synopsis refinement.

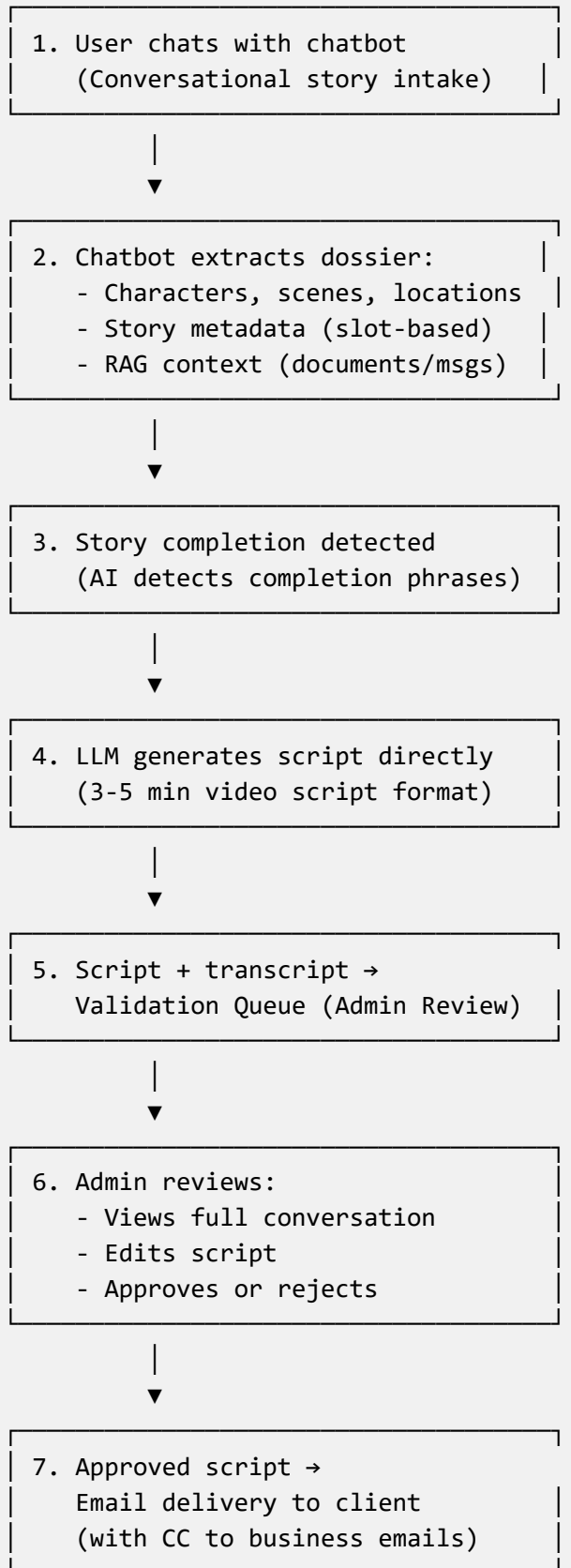
**Key Changes:**

-  User validation replaces admin validation
  -  Synopsis generation added as intermediate step
  -  Shot list replaces script as final output
  -  Iteration loop for synopsis refinement
  -  RAG usage becomes more explicit and critical
- 

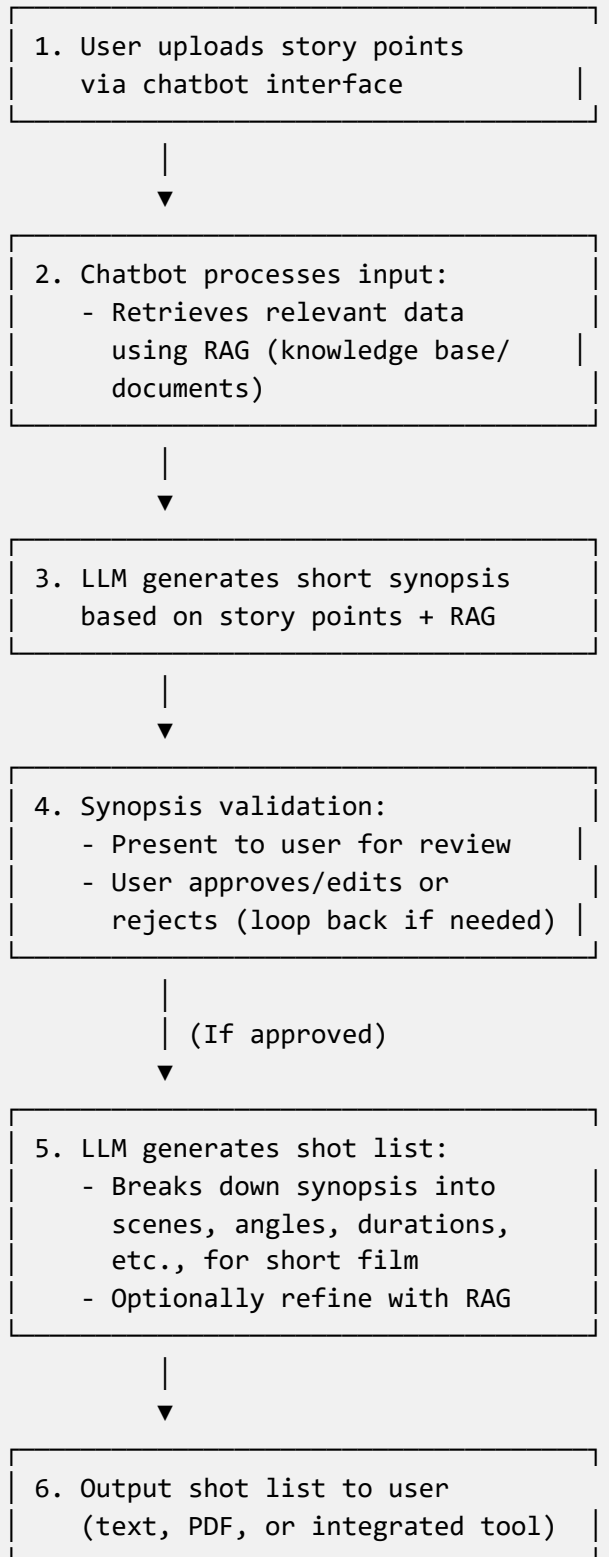
## Workflow Comparison

---

### Current Flow (Existing Implementation)



## New Requirements Flow (Client Request)



## Detailed Differences Analysis

### 1. Input Method

Aspect	Current Flow	New Requirements
--------	--------------	------------------

Aspect	Current Flow	New Requirements
<b>Input Type</b>	Conversational chat (Q&A format)	Story points upload (bulk input)
<b>Interaction Style</b>	Interactive, guided questions	Batch upload, then processing
<b>Data Collection</b>	Slot-based extraction during conversation	Pre-structured story points
<b>User Experience</b>	Step-by-step guidance	Upload → Process → Review

**Impact:** Significant change in UX. Current system is conversational; new system expects structured input upfront.

## 2. RAG Usage

Aspect	Current Flow	New Requirements
<b>RAG Timing</b>	Used during conversation for context	Explicitly used after input processing
<b>RAG Purpose</b>	Enhance conversation context, document analysis	Retrieve relevant knowledge base data
<b>RAG Visibility</b>	Implicit (background enhancement)	Explicit (primary processing step)
<b>RAG Integration</b>	Integrated into chat responses	Separate processing step

**Impact:** RAG becomes more critical and visible. May need enhanced RAG capabilities for knowledge base retrieval.

## 3. Synopsis Generation

Aspect	Current Flow	New Requirements
<b>Synopsis Step</b>	✗ Not implemented (planned but not done)	✓ Required as intermediate step
<b>Timing</b>	Would be after story completion	After story points processing
<b>Purpose</b>	Summary for admin review	User validation before shot list
<b>Output Format</b>	N/A	Short synopsis (1-2 pages?)

**Impact:** New feature that needs to be built. Currently synopsis generation is 90% planned but not implemented.

## 4. Validation Workflow

Aspect	Current Flow	New Requirements
Validator	Admin (internal team)	User (client)
Validation Target	Script	Synopsis
Validation Location	Admin dashboard (/admin)	User interface (chatbot)
Iteration Loop	Single review → approve/reject	Loop back to step 3 if rejected
Feedback Mechanism	Admin notes in validation queue	User edits/rejects with feedback

**Impact:** Major architectural change. Shifts from admin-controlled to user-controlled validation.

## 5. Output Type

Aspect	Current Flow	New Requirements
Final Output	Video script (3-5 min format)	Shot list (scenes, angles, durations)
Output Format	Text script	Structured shot list (text/PDF/tool)
Output Recipient	Client via email	User directly in interface
Output Timing	After admin approval	After user synopsis approval

**Impact:** Different output format and delivery method. Shot list is more structured than script.


## 6. Iteration & Feedback Loop

Aspect	Current Flow	New Requirements
Iteration	Single pass (admin can edit)	Multiple passes (user can loop)
Loop Trigger	Admin rejection	User rejection of synopsis
Loop Destination	Back to admin for re-review	Back to synopsis generation (step 3)
Feedback Integration	Admin notes	User edits/feedback in prompt

**Impact:** Requires state management for iteration cycles and feedback incorporation.

# Implementation Requirements

## 1. Story Points Upload Interface

**Current State:**  Chat interface exists, but designed for conversational input

**Required Changes:**


- Add bulk upload capability for story points

- Support text file upload or structured input form
- Parse and structure uploaded story points

**Complexity:** Medium

- Need to handle file uploads (already supported for images/documents)
  - Need to parse and structure story points
  - May need validation for incomplete/invalid story points
- 

## 2. Enhanced RAG Processing

**Current State:**  RAG exists but used implicitly during conversation

**Required Changes:**

- Make RAG processing explicit and visible
- Enhance RAG to retrieve from knowledge base (not just documents)
- Display RAG context to user (optional, for transparency)

**Complexity:** Medium-High

- RAG infrastructure exists but may need enhancement
  - Knowledge base integration may be new
  - Need to ensure RAG context is relevant and useful
- 

## 3. Synopsis Generation

**Current State:**  Planned but not implemented (90% complete per proposal)

**Required Changes:**

- Implement synopsis generation LLM call
- Generate synopsis from story points + RAG context
- Format synopsis appropriately (1-2 pages?)

**Complexity:** Low-Medium

- LLM infrastructure exists
  - Need to create synopsis generation prompt
  - Need to store synopsis in database
- 

## 4. User Validation Interface

**Current State:**  Admin validation exists, user validation does not

**Required Changes:**

- Create user-facing validation UI in chatbot
- Display synopsis to user
- Allow user to approve/edit/reject
- Capture user feedback for iteration

**Complexity:** High

- New UI component needed
  - State management for validation status
  - Integration with chatbot interface
  - User feedback capture and storage
- 

## 5. Iteration Loop Logic

**Current State:** ❌ No iteration loop exists

**Required Changes:**

- Detect user rejection
- Loop back to synopsis generation
- Incorporate user feedback into LLM prompt
- Track iteration count (prevent infinite loops)

**Complexity:** Medium-High

- State machine for workflow stages
  - Feedback incorporation logic
  - Loop detection and limits
  - User experience for iteration
- 

## 6. Shot List Generation

**Current State:** ❌ Script generation exists, shot list does not

**Required Changes:**

- Create shot list generation LLM call
- Generate structured shot list (scenes, angles, durations)
- Format shot list appropriately
- Support multiple output formats (text/PDF/tool)

**Complexity:** Medium

- Similar to script generation (can reuse patterns)
  - Different output structure (shot list vs. script)
  - May need structured output format (JSON?)
- 

## 7. Output Delivery

**Current State:** ✅ Email delivery exists for scripts

**Required Changes:**

- Deliver shot list directly to user interface
- Support multiple formats (text, PDF, tool integration)
- Remove email delivery (or make optional)

**Complexity:** Low-Medium

- UI display is straightforward
  - PDF generation may need new library
  - Tool integration depends on target tools
- 

## Frontend Changes Required

---

### New Components

#### 1. Story Points Upload Component

- File upload or text input
- Preview uploaded content
- Validation feedback

#### 2. Synopsis Display Component

- Show generated synopsis
- Edit interface
- Approve/Reject buttons
- Feedback input

#### 3. Shot List Display Component

- Structured shot list view
- Export options (PDF, text)
- Integration with film tools (future)

### Modified Components

#### 1. ChatPanel

- Support story points upload mode
- Display synopsis validation UI
- Show iteration status

#### 2. SessionsSidebar

- Show synopsis status
  - Show shot list status
  - Filter by workflow stage
- 

## Complexity Assessment

---

### Low Complexity

- Shot list generation (similar to script generation)
- Output delivery (UI display)
- Database schema additions (straightforward fields)



## Medium Complexity ⚠️

- Story points upload and parsing
- Synopsis generation (LLM call similar to script)
- Enhanced RAG processing
- Iteration loop logic

## High Complexity ●

- User validation interface (new UX paradigm)
  - State management for workflow stages
  - Feedback incorporation into LLM prompts
  - Workflow state machine
- 

## Timeline Estimate

---

### Phase 1: Core Infrastructure (3-4 days)

- Story points upload interface
- Enhanced RAG processing
- Synopsis generation
- Database schema updates

### Phase 2: User Validation (1-2 days)

- User validation UI
- Feedback capture
- State management

### Phase 3: Iteration Loop (1-2 days)

- Loop logic implementation
- Feedback incorporation
- Iteration limits and handling

### Phase 4: Shot List Generation (2-3 days)

- Shot list LLM generation
- Structured output format
- Display and export

### Phase 5: Testing & Refinement (4-5 days)

- End-to-end testing
- UX refinement
- Error handling
- Performance optimization

## Questions for Client

---

### 1. Workflow Compatibility

- Should we maintain the current conversational flow as an option?
- Or completely replace it with story points upload?

### 2. Admin Review

- Should admin review be removed entirely?
- Or kept as optional quality check after user validation?

### 3. Script vs. Shot List

- Should we generate both script and shot list?
- Or replace script entirely with shot list?

### 4. Synopsis Format

- What length/format for synopsis? (1-2 pages? Bullet points?)
- Should synopsis be editable by user or regenerated?

### 5. Iteration Limits

- Maximum number of iterations allowed?
- What happens after max iterations reached?

### 6. Output Format

- Preferred shot list format? (JSON, text, PDF, tool-specific?)
- Which film tools should we integrate with?

### 7. RAG Knowledge Base

- What knowledge base should RAG query?
  - Existing documents or new knowledge base?
- 

## Recommendations

---

### 1. Phased Implementation

- Start with story points upload + synopsis generation
- Add user validation in phase 2
- Add shot list generation in phase 3
- Refine iteration loop in phase 4

### 2. Maintain Backward Compatibility

- Keep current conversational flow as option
- Allow users to choose workflow style

- Gradual migration path

### 3. Hybrid Approach

- Use admin review as fallback for rejected synopses
- Allow admin to intervene if user iterations exceed limit
- Maintain quality control layer

### 4. Enhanced RAG

- Invest in RAG quality improvements
- Build knowledge base if needed
- Monitor RAG performance metrics

### 5. User Education

- Clear onboarding for new workflow
  - Help documentation
  - Tooltips and guidance throughout
- 

## Conclusion

---

The new workflow requirements represent a **significant shift** from the current implementation:

- **User-driven** instead of admin-driven
- **Two-stage generation** (synopsis → shot list) instead of single script
- **Iteration loop** for refinement
- **Different output format** (shot list vs. script)

#### Key Challenges:

1. User validation UX design
2. Iteration loop state management
3. RAG quality and relevance
4. Workflow state persistence

**Recommendation:** Proceed with phased implementation, starting with core infrastructure and gradually adding user validation and iteration features.

---