Prompts used for both building the tool itself and prompts used within the tool

Prompts used for Building the tool:

1) "I want to make an AI storytelling tool which takes a feature description and converts it into an 8-chunk video story. Can you give me the basic plan …like what classes I should create, how Perplexity will do analysis, how … VEO3 will generate the videos.."

2) "Tell me how the VEO3 credit are free... and how many video generations will one credit give and how much does each video chunk cost?"

3) "I want to calculate how many VEO3 credits I will need for my project. Each project has 8 chunks 8s each...

4) "Give me the official limits for perplexity API.. like tokens per request, daily cap, and how credits are counted. I want it in a simple table format."

5) "Sometimes I get errors because of token exhusted in Perplexity..write a helper prompt that will auto-summarize my input feature description to fit inside the allowed token count."

6) "Research if VEO3 has different credit pricing for fast-preview vs high-quality. I want to know which one to use when I'm testing vs final run…and which are the models available the 300$ credit.

7) "Give me proper JSON formats for things like feature details, analysis report, narrative plan, chunk info, and QA report. I should be able to use them in my code directly. Add examples also."

8) "Write me a good prompt for Perplexity which can take a feature description and break it into clear insights like name, category, value points, pain points, and use cases. I want JSON output only."

9) "Help me map the analysis into an 8-part emotional story arc. I want proper things like struggle, escalation, discovery, trial, breakthrough, mastery, success, future planning. Each part should have emotion, setting, and what feature aspect is shown."

10) "Create a function idea where each chunk becomes a proper scene spec. It should have environment, action, camera, characters, and timing. Basically one JSON block per chunk."

11) "I need a small library of reusable prompts….like for feature analysis, narrative building, scene director, qa, and doc writer… just give me clear text templates which I can plug inside code later."

12) "Show me how to pack a scene into a VEO3 generation prompt. It should include setting, camera, actions and a negative prompt so that no wrong things like showing camera, subtitles, etc. happen."

13) "Help me design a prompt that creates a short voiceover narration for each chunk. It should be punchy and finish within 8s. Give example lines also."

14) "Make a simple subtitle generator prompt which will turn the narration into SRT format. Each chunk should have 8s slot and text length should not overflow."

15) "Give me a qa checking prompt which will score my output on story flow, emotional effect, and VEO3 rules. I want it to say where it fails and also what to fix."

16) "Sometimes the video gen is getting blocked... write me a retry prompt which will auto-rewrite sensitive lines."

17) "I want to write a small style guide prompt. It should say what kind of visuals, tone, color words and character design I have to use and also add what not to do like don't show camera, don't use text on screen."

18) "Make a localized ready prompt so that if I need to adapt the story to other languages, it is easy and add rules also to avoid some non-understandable terms."

19) "Sometimes the feature description will be too short. I need a clarification prompt which can ask 4-5 questions to user to fill the gaps. If no answers, then fallback defaults should be used."

20) "Write a prompt idea for logging. It should record timestamp, inputs, outputs, retry info, and QA score

21) "I want my outputs to be consistent. Write a prompt that ensures deterministic generation by using same seeds, memoization, and chunk caching. Also say when to clear cache."

22) "Design prompts for the user flow text. Like what the tool should show when uploading, previewing, choosing narrative tone, confirming, generating, and exporting. Add small microcopy lines like 'video will be 52 seconds in total'."

23) "Make me an end to end prompt. It should call analysis, story arc, scenes, VO, subtitles, VEO3 prompts, QA, and final doc. Also ask it to check all chunks are exactly 8s "

Prompts Used within the tool

Feature analysis prompt:

```
analysis_prompt = f"""
You are an expert product analyst specializing in feature breakdown
and stakeholder mapping.

Analyze this DETAILED FEATURE DESCRIPTION and extract comprehensive
insights:

FEATURE DESCRIPTION:
{feature_description}

EXTRACT THE FOLLOWING INFORMATION (JSON format):
{{
    "feature_name": "Extract the feature name",
    "feature_category": "Classify the feature type (e.g., 'User
Engagement Tool', 'Analytics Platform', etc.)",
    "core_value_propositions": [
        "Primary value proposition 1",
        "Primary value proposition 2",
        "Primary value proposition 3"
    ],
    "customer_pain_points": [
        "Specific pain point this feature solves 1",
        "Specific pain point this feature solves 2",
        "Specific pain point this feature solves 3",
        "Specific pain point this feature solves 4"
    ],
    "key_capabilities": [
        "Technical capability 1",
        "Technical capability 2",
        "Technical capability 3"
    ],
    "quantified_benefits": [
        "Measurable benefit 1 (include numbers if provided)",
        "Measurable benefit 2 (include numbers if provided)",
        "Measurable benefit 3 (include numbers if provided)"
    ],
    "target_user_personas": [
        {{
            "persona_name": "Primary User Type",
            "role": "Their job role/position",
            "current_challenges": "What they struggle with currently",
            "desired_outcomes": "What success looks like for them",
            "usage_context": "When/where they use this feature"
```

```
                }},
                {{
                        "persona_name": "Secondary User Type",
                        "role": "Their job role/position",
                        "current_challenges": "What they struggle with currently",
                        "desired_outcomes": "What success looks like for them",
                        "usage_context": "When/where they use this feature"
                }}
        ],
        "stakeholder_journeys": {{
                "primary_user": {{
                        "persona": "Primary user description with role",
                        "current_state": "How they currently handle this
problem/need",
                        "pain_points": ["Current frustration 1", "Current
frustration 2"],
                        "discovery_trigger": "What makes them look for a
solution",
                        "evaluation_criteria": ["What they care about when
choosing", "Success metric 1", "Success metric 2"],
                        "adoption_barriers": ["Potential concern 1", "Potential
concern 2"],
                        "desired_state": "Their ideal outcome with this feature",
                        "success_indicators": ["How they measure success 1", "How
they measure success 2"],
                        "emotional_arc": "Frustration → Hope → Confidence →
Success"
                }},
                "secondary_stakeholders": [
                        {{
                                "role": "Secondary stakeholder role",
                                "relationship_to_primary": "How they relate to primary
user",
                                "impact": "How this feature affects them",
                                "concerns": "Their potential concerns or needs",
                                "success_criteria": "What success means for them"
                        }}
                ]
        }},
        "use_cases": [
                {{
                        "scenario": "Use case scenario 1",
                        "context": "When/where this happens",
                        "user_goal": "What user wants to achieve",
                        "feature_role": "How feature helps achieve goal",
                        "outcome": "Expected result"
                }},
                {{
```

```
                "scenario": "Use case scenario 2",
                "context": "When/where this happens",
                "user_goal": "What user wants to achieve",
                "feature_role": "How feature helps achieve goal",
                "outcome": "Expected result"
            }},
            {{
                "scenario": "Use case scenario 3",
                "context": "When/where this happens",
                "user_goal": "What user wants to achieve",
                "feature_role": "How feature helps achieve goal",
                "outcome": "Expected result"
            }}
        ],
        "business_impact": {{
            "efficiency_gains": "How this improves efficiency (with
numbers if available)",
            "cost_savings": "Potential cost reductions (with numbers if
available)",
            "revenue_opportunities": "Revenue impact (with numbers if
available)",
            "competitive_advantages": "Market differentiation points",
            "roi_indicators": ["ROI metric 1", "ROI metric 2", "ROI metric
3"]
        }},
        "technical_requirements": [
            "Key technical requirement 1",
            "Key technical requirement 2",
            "Integration requirement 1",
            "Performance requirement 1"
        ],
        "success_metrics": [
            "Quantifiable outcome metric 1",
            "User behavior metric 1",
            "Business impact metric 1",
            "Adoption metric 1"
        ],
        "narrative_themes": {{
            "central_conflict": "Main problem/challenge this feature
addresses",
            "resolution_path": "How feature resolves the conflict",
            "transformation_story": "The before/after user
transformation",
            "emotional_beats": [
                "Current frustration/struggle",
                "Problem escalation/pain",
                "Discovery of solution",
                "Initial hope/curiosity",
```

```
                "Engagement/trial",
                "Growing confidence",
                "Transformation/success",
                "Mastery/celebration"
            ],
            "visual_metaphors": [
                "Metaphor representing current state",
                "Metaphor representing transformation",
                "Metaphor representing success state"
            ],
            "story_settings": [
                "Real-world environment 1 where feature is used",
                "Real-world environment 2 where feature is used",
                "Real-world environment 3 where feature is used"
            ]
        }},
        "competitive_differentiation": {{
            "unique_advantages": ["What makes this feature different 1",
"What makes this feature different 2"],
            "market_gaps": ["Gap this fills 1", "Gap this fills 2"],
            "user_preference_drivers": ["Why users choose this 1", "Why
users choose this 2"]
        }}
    }}

    IMPORTANT: Provide ONLY the JSON response. Extract specific details
from the feature description provided. If quantified benefits or metrics are
mentioned, include them exactly. If use cases are described, extract them
precisely.
        """
```

Enhanced Story Analysis:

```
    enhanced_story_prompt = f"""
You are creating a DRAMATIC FEATURE TRANSFORMATION STORY that follows a
complete emotional journey from devastation to triumph, while showcasing the
complete value and transformation delivered by this feature.

==== MANDATORY 8-CHUNK DRAMATIC STORY ARC ====

CHUNK 1: **MANUAL STRUGGLE ACTIVITY** - Character doing frustrating manual
work (filing papers, manual data entry, phone calls, physical sorting)

CHUNK 2: **CRISIS ESCALATION ACTIVITY** - Character dealing with urgent
problems (emergency meeting, rushing with documents, fixing broken process)
```

CHUNK 3: **DISCOVERY ACTIVITY** - Character researching solutions (reading, browsing, asking colleagues, investigating options)

CHUNK 4: **TRIAL ACTIVITY** - Character cautiously testing new approach (setup, configuration, first attempt, careful observation)

CHUNK 5: **BREAKTHROUGH ACTIVITY** - Character experiencing success (celebrating, sharing results, demonstrating to others)

CHUNK 6: **MASTERY ACTIVITY** - Character becoming expert (training others, optimizing, scaling up usage)

CHUNK 7: **SUCCESS DEMONSTRATION** - Character presenting achievements (client meeting, presentation, showing results)

CHUNK 8: **FUTURE PLANNING** - Character planning expansion (strategy session, road mapping, mentoring others)

==== ACTIVITY VARIETY ENFORCEMENT ====

CRITICAL: Each chunk must show COMPLETELY DIFFERENT professional activities:
- NO repetitive dashboard viewing across multiple chunks
- NO similar screen-checking activities
- Each scene shows UNIQUE work scenarios and interactions
- Vary between: meetings, presentations, hands-on work, collaboration, planning, problem-solving
- Different locations: office, conference room, workshop, client site, cafe, home office

==== FEATURE ANALYSIS INTEGRATION ====

FEATURE NAME: {feature_name}
CATEGORY: {feature_analysis.get('feature_category', 'Business Tool')}

CORE VALUE PROPOSITIONS:
{chr(10).join(f"• {value}" for value in core_values[:3])}

CUSTOMER PAIN POINTS TO DEMONSTRATE:
{chr(10).join(f"• {pain}" for pain in pain_points[:4])}

KEY CAPABILITIES TO SHOWCASE:
{chr(10).join(f"• {cap}" for cap in capabilities[:3])}

QUANTIFIED BENEFITS TO HIGHLIGHT:
{chr(10).join(f"• {benefit}" for benefit in benefits[:3])}

==== STAKEHOLDER JOURNEY INTEGRATION ====

```
PRIMARY USER PERSONA: {stakeholder_journey.get('persona', 'Professional
user')}
CURRENT STATE: {stakeholder_journey.get('current_state', 'Manual processes')}
PAIN POINTS: {', '.join(stakeholder_journey.get('pain_points', []))[:2])}
DESIRED STATE: {stakeholder_journey.get('desired_state', 'Efficient automated
workflow')}
SUCCESS INDICATORS: {', '.join(stakeholder_journey.get('success_indicators',
[]))[:2])}
EMOTIONAL ARC: {stakeholder_journey.get('emotional_arc', 'Challenge →
Success')}

==== USE CASE INTEGRATION ====
{chr(10).join(f"USE CASE {i+1}: {case.get('scenario', 'N/A')} -
{case.get('outcome', 'N/A')}" for i, case in enumerate(use_cases[:3]))}

==== NARRATIVE THEMES ====
CENTRAL CONFLICT: {narrative_themes.get('central_conflict', 'User challenge')}
RESOLUTION PATH: {narrative_themes.get('resolution_path', 'Feature solution')}
TRANSFORMATION STORY: {narrative_themes.get('transformation_story',
'Before/after improvement')}
VISUAL METAPHORS: {', '.join(narrative_themes.get('visual_metaphors',
[]))[:3])}
STORY SETTINGS: {', '.join(narrative_themes.get('story_settings', []))[:3])}

==== EMOTIONAL PROGRESSION MAP ====
{json.dumps(emotional_progression, indent=2)}

==== CINEMATIC STORY REQUIREMENTS ====

1. **FEATURE VALUE DEMONSTRATION**: Each chunk must progressively demonstrate
specific feature capabilities and their impact on real user pain points.

2. **STAKEHOLDER JOURNEY ALIGNMENT**: Follow the primary user's complete
transformation from current frustrating state to desired successful state.

3. **USE CASE INTEGRATION**: Incorporate real use cases from the feature
analysis into scene contexts.

4. **EMOTIONAL ARC MASTERY**: Hit precise emotional beats that build authentic
engagement and show genuine transformation.

5. **QUANTIFIED IMPACT SHOWCASE**: Where possible, visually represent the
quantified benefits and improvements.

6. **PROGRESSIVE CAPABILITY REVEAL**: Each scene should reveal new aspects of
the feature's capabilities and value.
```

7. **AUTHENTIC PROFESSIONAL CONTEXT**: Show the feature in realistic professional environments and workflows.

8. **DRAMATIC EMOTIONAL STORYTELLING**: Each chunk must show authentic human emotion and personal stakes, making viewers feel the character's journey from devastation to triumph.

9. **ACTIVITY DIVERSITY MANDATE**: Each chunk must demonstrate the feature through COMPLETELY different professional activities and scenarios:
   - Chunk 1: Manual/traditional work methods
   - Chunk 2: Crisis management/urgent situations
   - Chunk 3: Research/discovery activities
   - Chunk 4: Initial testing/setup activities
   - Chunk 5: Active feature usage/breakthrough moments
   - Chunk 6: Advanced usage/optimization activities
   - Chunk 7: Results presentation/demonstration activities
   - Chunk 8: Strategic planning/mentoring activities

10. **VISUAL VARIETY REQUIREMENTS**: Ensure each scene shows different:
    - Physical activities (typing vs. presenting vs. collaborating vs. analyzing)
    - Work tools (computers vs. whiteboards vs. documents vs. mobile devices)
    - Professional interactions (solo work vs. team meetings vs. client presentations)
    - Environmental settings (office desk vs. conference room vs. workshop vs. client site)

==== ACTIVITY EXAMPLES FOR EACH CHUNK ====

CHUNK 1 Examples:
- Manually sorting through physical feedback forms
- Making individual phone calls to collect opinions
- Typing survey responses into spreadsheets by hand
- Searching through email chains for feedback

CHUNK 2 Examples:
- Rushing to prepare last-minute reports
- Emergency meeting due to lack of data
- Frantically calling clients for missing information
- Working late trying to compile manual reports

CHUNK 3 Examples:
- Reading articles about feedback solutions
- Having coffee conversation with colleague about tools
- Browsing software comparison websites
- Attending demo or webinar about new solutions

CHUNK 4 Examples:

- Setting up software for first time
- Creating first survey or feedback form
- Training team members on new process
- Testing with small group or pilot project

CHUNK 5 Examples:
- Receiving first real-time feedback notifications
- Watching live response analytics come in
- Sharing exciting results with team members
- Celebrating first successful campaign

CHUNK 6 Examples:
- Optimizing advanced features and settings
- Creating sophisticated automated workflows
- Training other departments on the system
- Scaling successful processes company-wide

CHUNK 7 Examples:
- Presenting impressive results to executives
- Demonstrating ROI to stakeholders
- Showing before/after comparisons to clients
- Receiving recognition or awards for improvements

CHUNK 8 Examples:
- Planning future feature implementations
- Mentoring new team members on best practices
- Discussing integration with other business tools
- Strategizing about expanding usage across organization

==== FORMAT FOR EACH CHUNK (8 TOTAL) ====

CHUNK X: [Emotional Story Title + Feature Capability - e.g., "The Breaking Point - Feedback Collection Crisis"]

CHARACTER DNA: [IDENTICAL character descriptions representing primary user persona - MUST BE EXACT SAME IN ALL CHUNKS]

ACTIVITY VARIATION REQUIREMENT: While the character remains identical, they must be shown in COMPLETELY different professional activities across the 8 chunks. Same person, different work scenarios and tasks in each scene.

[Include: Role, appearance, clothing, mannerisms, background - COPY EXACTLY TO ALL CHUNKS]
[MANDATE: Show this person doing VARIED professional activities - never repeat the same type of work task]

Primary User Persona: {stakeholder_journey.get('persona', 'Professional seeking efficiency')}
[Include: Role, appearance, clothing, mannerisms, background - COPY EXACTLY TO ALL CHUNKS]

[STAKEHOLDER CONTEXT]: [Current journey stage from emotional progression]

[PAIN POINT ADDRESSED]: [Specific pain point from analysis that this chunk addresses]

[FEATURE CAPABILITY]: [Specific capability being demonstrated in this scene]

[QUANTIFIED BENEFIT]: [Measurable benefit shown/implied in this scene]

[EMOTIONAL BEAT]: [Current emotional state with intensity level from progression map]

[USE CASE CONTEXT]: [Which use case scenario this relates to, if applicable]

[SETTING]: [UNIQUE professional environment from story settings - different from all other chunks - should reflect emotional state]

[CINEMATIC ACTION STORY]: [COMPLETE EMOTIONAL NARRATIVE SCENE - Write as compelling dramatic story with:
• Character's internal emotional state and external situation
• Rising dramatic tension or discovery moment
• Feature interaction/implementation within the emotional context
• Multiple connected actions showing both workflow and emotional journey
• Visible transformation/results and character's emotional response
• Scene conclusion with emotional impact and setup for next chunk
Write as engaging narrative prose - minimum 5-6 sentences telling the complete emotional and professional story of this moment. Show both the feature capabilities AND the human drama.]

[CAMERA]: {EXACT_CAMERA_TEXT}

[TIMING]: Duration: 6.5 seconds (5.5s movement + 1s concealment behind right object)

[VISUAL FEATURE DEMONSTRATION]: [How the feature's value is visually communicated without on-screen text]

[PROFESSIONAL CONTEXT]: [Realistic workplace/professional setting details]

[TRANSFORMATION ELEMENT]: [What specific improvement/change is visible in this scene]

[AUDIO]: [Professional ambient sounds, any natural dialogue that reinforces feature benefits and emotional state]

[VISUAL METAPHORS]: [Incorporate relevant metaphors from analysis that support emotional journey]

[STAKEHOLDER SUCCESS INDICATOR]: [How success is visually represented for this user persona]

[NARRATIVE PROGRESSION]: [How this chunk advances the complete feature demonstration story and emotional arc]

[EXCLUSIONS]: No on-screen text, captions, subtitles, characters never look at camera, no footage after concealment

==== SPECIFIC DRAMATIC STORY PROGRESSION ====

CHUNK 1: Character faces complete professional failure with current methods, devastated about career prospects, shows failed feedback collection attempts
CHUNK 2: Crisis escalates, everything falls apart, character questions abilities, manual processes completely break down
CHUNK 3: In desperation, character searches for solutions and discovers {feature_name} - first glimmer of hope
CHUNK 4: Character cautiously tries the feature, experiences small wins with real-time feedback, dares to hope
CHUNK 5: Major breakthrough! Feature works brilliantly, character amazed by instant engagement and data flow
CHUNK 6: Character gains mastery and confidence, sees professional life transforming through advanced capabilities
CHUNK 7: Complete success achieved, character celebrates transformation with comprehensive analytics and recognition
CHUNK 8: Character becomes mentor/advocate, discusses future enhancements and inspires others

==== CRITICAL SUCCESS REQUIREMENTS ====

- Show REAL professional scenarios where this feature delivers value
- Demonstrate ACTUAL capabilities from the feature analysis
- Follow AUTHENTIC stakeholder journey progression
- Include SPECIFIC use case contexts from analysis
- Show MEASURABLE improvements and transformation
- Maintain IDENTICAL character DNA (same person throughout story)
- Each scene shows COMPLETELY DIFFERENT professional environment
- Feature benefits must be VISUALLY CLEAR without text overlays
- Emotional progression must feel GENUINE and earned with real human stakes
- Show the character's internal thoughts, fears, hopes, and dreams
- Use environmental details to reflect emotional states
- Create genuine dramatic tension and satisfying emotional resolution

```
- The feature should feel like a life-changing discovery, not just a tool
- Precise 6.5-second timing with perfect camera concealment

GENERATE 8 CONNECTED CHUNKS that tell the complete story of:
A devastated professional discovering "{feature_name}" and achieving life-
changing transformation from crushing failure to triumphant success and future
vision.

Each chunk must demonstrate different aspects of the feature while showing the
same character's emotional and professional journey through different
environments.

"""
```

Master prompt:

```python
        master_prompt = f"""
MASTER VEO3 AGENT PROMPT - FEATURE-DRIVEN CINEMATIC STORYTELLING

==== FEATURE ANALYSIS FOUNDATION ====
Feature Name: {feature_analysis.get('feature_name', 'N/A')}
Feature Category: {feature_analysis.get('feature_category', 'N/A')}
Core Value Propositions:
{json.dumps(feature_analysis.get('core_value_propositions', []), indent=2)}
Customer Pain Points: {json.dumps(feature_analysis.get('customer_pain_points',
[]), indent=2)}
Key Capabilities: {json.dumps(feature_analysis.get('key_capabilities', []),
indent=2)}
Quantified Benefits: {json.dumps(feature_analysis.get('quantified_benefits',
[]), indent=2)}

==== STAKEHOLDER JOURNEY MAPPING ====
Primary User Persona: {feature_analysis.get('stakeholder_journeys',
{}).get('primary_user', {}).get('persona', 'N/A')}
Current State: {feature_analysis.get('stakeholder_journeys',
{}).get('primary_user', {}).get('current_state', 'N/A')}
Pain Points: {json.dumps(feature_analysis.get('stakeholder_journeys',
{}).get('primary_user', {}).get('pain_points', []))}
Desired State: {feature_analysis.get('stakeholder_journeys',
{}).get('primary_user', {}).get('desired_state', 'N/A')}
Success Indicators: {json.dumps(feature_analysis.get('stakeholder_journeys',
{}).get('primary_user', {}).get('success_indicators', []))}
Emotional Arc: {feature_analysis.get('stakeholder_journeys',
{}).get('primary_user', {}).get('emotional_arc', 'N/A')}

==== USE CASE INTEGRATION ====
{json.dumps(feature_analysis.get('use_cases', []), indent=2)}

==== NARRATIVE EXCELLENCE FRAMEWORK ====
```

```
Central Conflict: {feature_analysis.get('narrative_themes',
{}).get('central_conflict', 'N/A')}
Resolution Path: {feature_analysis.get('narrative_themes',
{}).get('resolution_path', 'N/A')}
Transformation Story: {feature_analysis.get('narrative_themes',
{}).get('transformation_story', 'N/A')}
Emotional Beats: {json.dumps(feature_analysis.get('narrative_themes',
{}).get('emotional_beats', []))}
Visual Metaphors: {json.dumps(feature_analysis.get('narrative_themes',
{}).get('visual_metaphors', []))}
Story Settings: {json.dumps(feature_analysis.get('narrative_themes',
{}).get('story_settings', []))}

==== 8-CHUNK EMOTIONAL PROGRESSION ====
{json.dumps(narrative_structure.get('emotional_progression', {}), indent=2)}

==== VEO3 TECHNICAL SPECIFICATIONS ====
1. PRECISE TIMING CONTROL:
   - Total Duration: EXACTLY 6.5 seconds per chunk
   - Camera Movement: 5.5 seconds lateral dolly movement
   - Concealment Phase: 1.0 seconds behind right object
   - NO footage after concealment point

2. CHARACTER CONSISTENCY PROTOCOL:
   - Identical character DNA across all 8 chunks
   - Same facial features, clothing, and physical characteristics
   - Consistent behavior patterns aligned with stakeholder persona
   - Professional appearance matching target user persona

3. SCENE DIVERSITY WITH PURPOSE:
   - Each chunk shows COMPLETELY different professional environment
   - Progressive feature demonstration through varied contexts
   - Visual variety supporting emotional progression and use case integration

4. CAMERA MOVEMENT SPECIFICATION:
   {EXACT_CAMERA_TEXT}

5. FEATURE DEMONSTRATION REQUIREMENTS:
   - Visual communication of feature value without on-screen text
   - Authentic professional scenarios from use case analysis
   - Clear capability demonstration in realistic contexts
   - Transformation elements showing before/after improvements

==== QUALITY ASSURANCE FRAMEWORK ====
- Feature value proposition alignment verification
- Stakeholder journey progression validation
- Use case integration authenticity check
- Emotional engagement escalation confirmation
```

```
- VEO3 technical specification compliance
- Character consistency across all chunks
- Professional scenario authenticity validation
- Quantified benefit visual representation assessment

==== SUCCESS CRITERIA ====
- Complete feature story told across 8 connected chunks
- Authentic stakeholder transformation demonstrated
- Real use cases integrated into narrative
- Professional quality cinematic presentation
- Precise timing with perfect camera concealment
- Identical characters throughout different professional scenarios
- Clear feature value communication without text overlays
"""
```

VE03 Promp:

```
    enhanced_veo3_prompt = f"""

FEATURE-DRIVEN CINEMATIC CHUNK {chunk_index}/8 - PRECISE 6.5-SECOND TIMING
WITH VOICE-OVER

=== CHARACTER CONSISTENCY (EXACT DNA) ===
{character_dna}

=== FEATURE DEMONSTRATION CONTEXT ===
Core Feature Capability: {feature_capability}
Quantified Benefit: {quantified_benefit}
Emotional State: {emotional_state} (Intensity:
{emotional_beat.get('intensity', 5)}/10)
Narrative Beat: {emotional_beat.get('beat', 'Story development')}

=== UNIQUE SCENE SPECIFICATION ===
Setting: {setting}
Action: {action}

=== VOICE-OVER AUDIO INTEGRATION ===
**Cinematic Narrator Script (6.5 seconds):**
"{voiceover_script}"

**Audio Requirements:**
- - Primary Audio: Cinematic narrator voice-over (provided script above)
- **CRITICAL**: Use identical narrator voice across ALL video chunks -
consistent male documentary narrator (Morgan Freeman style)
- Secondary Audio: Natural ambient sounds appropriate for {setting}
- Secondary Audio: Natural ambient sounds appropriate for {setting}
```

- Background Audio: Optional realistic conversations/ambient dialogue (NOT characters addressing camera)
- Audio Style: Professional cinematic quality, story-driven narrative tone
- Narrator Voice: Authoritative yet empathetic cinematic storyteller (NOT product demo voice)
- Audio Timing: Voice-over must align perfectly with 6.5-second visual duration

Prohibited Audio Elements:
- No on-screen text captions or subtitles
- No characters directly addressing the audience
- No product demo-style narration
- No technical jargon in dialogue
- Voice-over should not feel like advertising copy

=== CRITICAL TIMING REQUIREMENTS ===
Total Duration: EXACTLY 6.5 seconds
Phase 1 (0-5.5s): {EXACT_CAMERA_TEXT}
Phase 2 (5.5-6.5s): Camera becomes fully concealed behind right-side object
MANDATORY: Video ENDS exactly when camera is completely hidden
Audio Sync: Voice-over script must complete within the 6.5-second timeframe

=== STAKEHOLDER JOURNEY ALIGNMENT ===
User Persona: {feature_analysis.get('stakeholder_journeys', {}).get('primary_user', {}).get('persona', 'Professional user')}
Journey Stage: {emotional_beat.get('stakeholder_stage', 'Feature exploration')}
Pain Point Addressed: {feature_analysis.get('customer_pain_points', ['User challenge'])[0] if feature_analysis.get('customer_pain_points') else 'User challenge'}

=== ADVANCED CINEMATIC REQUIREMENTS ===
- Emotional Progression: Show {emotional_state} building to {emotional_beat.get('intensity', 5)}/10 intensity
- Visual Storytelling: Use environment and actions to demonstrate feature value (complementing voice-over)
- Professional Quality: Cinematic lighting, composition, and movement
- Character Behavior: Natural, authentic interactions that never acknowledge camera
- Feature Integration: Clear demonstration of {feature_capability} within scene
- Audio-Visual Harmony: Visual action should complement voice-over narrative without literal description

=== TECHNICAL SPECIFICATIONS ===
- Resolution: Professional quality suitable for business presentation
- Aspect Ratio: 16:9 widescreen format
- Camera Movement: Precise dolly movement from left to right with exact timing

```
- Character Consistency: EXACT same people, faces, clothing, mannerisms as
specified
- Scene Uniqueness: Completely different from all other 7 chunks
- Timing Precision: Must end exactly at 6.5 seconds when camera is concealed
- Audio Quality: Professional cinematic audio with clear voice-over narration

=== QUALITY CONTROLS ===
- NO on-screen text, captions, or subtitles
- Characters never look directly at camera
- NO footage continues after camera concealment
- Maintain consistent visual style across story
- Demonstrate clear feature value through action and environment
- Support emotional progression of overall narrative
- Voice-over must carry the story and emotional arc
- Audio complements visuals without describing them literally

PRIORITY: Create a professional, feature-demonstrating scene with integrated
cinematic voice-over narration that ends with precise camera concealment at
exactly 6.5 seconds while advancing the stakeholder journey and emotional arc
through both visual and audio storytelling.

**Voice-Over Script to Include:**
"{voiceover_script}"

"""
```

Voice Over Prompt:

```
    voiceover_prompt = f"""

You are a master cinematic narrator scriptwriter creating voice-over narration
for a feature story video.

CONTEXT:
- Chunk {chunk_index}/8 in cinematic feature story
- Feature: {feature_analysis.get('feature_name', 'Product Feature')}
- Setting: {setting}
- Action: {action}
- Pain Point: {pain_point}
- Capability: {capability}
- Benefit: {benefit}
- Emotional Beat: {current_beat}
- Emotion: {emotion} (Intensity: {intensity}/10)

CINEMATIC VOICE-OVER REQUIREMENTS:

**Purpose**: Carry the story since no on-screen text is allowed
**Style**: Cinematic narrator voice (NOT product demo or character speaking to
camera)
```

```
**Duration**: EXACTLY 6.5 seconds when spoken at natural cinematic pace
**Tone**: {emotion} with {intensity}/10 emotional intensity - storytelling,
not selling

**8-Beat Emotional Progression Guidelines**:
- Chunk 1-2 (Frustration/Escalation): Show struggle, mounting pressure,
current pain
- Chunk 3-4 (Discovery/Hope): Introduce possibility, emerging solution,
growing optimism
- Chunk 5-6 (Engagement/Confidence): Show active use, building trust, visible
improvement
- Chunk 7-8 (Transformation/Celebration): Demonstrate success, achievement,
mastery

**Narrative Integration Rules**:
- Must align with the emotional arc and highlight: pain points → discovery →
transformation → success
- Reinforce feature benefits in plain, cinematic storytelling language (not
technical jargon)
- Feel like a movie narrator, not a product demonstration
- Use story-driven language that complements visual action
- Should enhance what's shown visually without describing it literally

**Voice-Over Content Focus for Chunk {chunk_index}**:
Current Beat: {current_beat}
Story Element: {pain_point if chunk_index <= 2 else capability if chunk_index
<= 6 else benefit}

**Cinematic Language Examples**:
- Instead of "The feature provides instant feedback" → "In that moment,
clarity emerged"
- Instead of "Users can track sentiment" → "Understanding flowed like never
before"
- Instead of "55% increase in engagement" → "Connection sparked, then ignited"
- Instead of "The platform automates processes" → "What once felt impossible
became effortless"

Generate ONLY the voice-over script text (no directions, no timestamps, just
the cinematic narration).
The script should be exactly the right length for 6.5 seconds when read at
natural storytelling pace.

Example length reference: "When deadlines crush your spirit and manual work
steals your dreams, you search for something more."

"""
```