YTEMpire Week 1 Execution Plan

Week 1 Overview

Sprint Theme: Core Functionality & First Video Generation
Primary Goal: Generate first end-to-end video through the platform
Success Metric: 10 test videos generated with <$3 cost tracking
Team Sync: Daily standups at 9:00 AM, integration testing at 4:00 PM

Executive Leadership

Role: CEO/Founder

Task 1: Investor Update Preparation

Description: Prepare Week 1 progress report demonstrating MVP momentum and first video generation. **Steps**:

1. Compile metrics from first video generation (cost, time, quality)
2. Create investor deck showing 12-week roadmap progress
3. Record demo video of platform generating content
4. Schedule investor calls for feedback **Duration**: 3 hours **Dependencies**: First successful video generation **Deliverable**: Investor update deck with demo video **Priority**: P1

Task 2: Beta User Interview Schedule

Description: Begin qualitative research with potential beta users to validate approach. **Steps**:

1. Contact 20 potential beta users from target demographic
2. Schedule 30-minute discovery calls (aim for 10 confirmed)
3. Prepare interview script focusing on pain points
4. Create feedback tracking spreadsheet **Duration**: 4 hours **Dependencies**: Beta user recruitment plan from Week 0 **Deliverable**: 10 scheduled user interviews with interview guide **Priority**: P2

Task 3: Strategic Partnership Outreach

Description: Initiate conversations with potential API partners for better rates. **Steps**:

1. Draft outreach emails to OpenAI, ElevenLabs enterprise teams
2. Prepare volume projections for negotiation (10K+ videos/month)
3. Schedule partnership calls
4. Document pricing tiers and volume discounts **Duration**: 2 hours **Dependencies**: Cost analysis from Week 0 **Deliverable**: Partnership outreach tracker with initial responses **Priority**: P2

Role: Product Owner

Task 1: User Story Refinement Sprint

Description: Detail user stories for Week 2-3 development based on Week 0 learnings. **Steps**:

1. Break down channel setup wizard into 8-10 user stories
2. Define acceptance criteria for video generation workflow
3. Prioritize stories based on user value and dependencies
4. Estimate story points with development team **Duration**: 4 hours **Dependencies**: Week 0 feature prioritization **Deliverable**: Refined backlog with 20 detailed user stories **Priority**: P0

Task 2: Quality Metrics Definition

Description: Establish measurable quality standards for generated content. **Steps**:

1. Define minimum quality score threshold (0-100 scale)
2. Create rubric for script coherence, voice quality, video smoothness
3. Set benchmark metrics from competitor analysis
4. Document automated vs manual quality checks **Duration**: 3 hours **Dependencies**: AI team model evaluation framework **Deliverable**: Quality standards document with scoring rubric **Priority**: P1

Task 3: Onboarding Flow Prototype

Description: Create interactive prototype of user onboarding experience. **Steps**:

1. Design 10-step onboarding wizard in Figma
2. Add interactive transitions and micro-interactions
3. Include channel selection and niche recommendation flow
4. Create prototype testing plan **Duration**: 4 hours **Dependencies**: UI/UX wireframes from Week 0 **Deliverable**: Clickable Figma prototype of onboarding flow **Priority**: P2

Role: CTO/Technical Director

Task 1: First Video Generation Orchestration

Description: Coordinate all teams to achieve first automated video generation. **Steps**:

1. Verify all pipeline components are connected
2. Run end-to-end test with monitoring at each stage
3. Debug any integration issues in real-time
4. Document successful configuration **Duration**: 4 hours **Dependencies**: All Week 0 infrastructure tasks **Deliverable**: First successfully generated video with metrics **Priority**: P0

Task 2: Performance Baseline Establishment

Description: Measure and document system performance metrics for optimization tracking. **Steps**:

1. Benchmark API response times across all endpoints
2. Measure video generation pipeline stages (script: 30s, voice: 60s, etc.)
3. Document resource utilization (CPU, GPU, memory)
4. Create performance tracking dashboard **Duration**: 3 hours **Dependencies**: Monitoring stack from Week 0 **Deliverable**: Performance baseline report with targets **Priority**: P1

Task 3: Technical Debt Registry

Description: Begin tracking technical shortcuts taken for MVP speed. **Steps**:

1. Create technical debt log in JIRA
2. Categorize debt by impact and effort to resolve
3. Prioritize critical security and scalability items
4. Plan debt resolution for post-MVP **Duration**: 2 hours **Dependencies**: Week 1 development progress **Deliverable**: Technical debt registry with 10-15 items **Priority**: P2

Task 4: Architecture Review Session

Description: Conduct architecture review with all technical leads to identify gaps. **Steps**:

1. Review Week 0 architecture against Week 1 learnings
2. Identify bottlenecks in current design
3. Document required architectural changes
4. Update architecture diagrams **Duration**: 3 hours **Dependencies**: First video generation completion **Deliverable**: Updated architecture document v1.1 **Priority**: P1

Role: VP of AI

Task 1: Content Quality Optimization

Description: Fine-tune AI models based on first video generation results. **Steps**:

1. Analyze quality scores from first 10 videos
2. Adjust prompt templates for better coherence
3. Implement content filtering for policy compliance
4. A/B test different model parameters **Duration**: 4 hours **Dependencies**: First video batch generated **Deliverable**: Optimized prompt templates v2.0 **Priority**: P0

Task 2: Cost Optimization Implementation

Description: Implement strategies to ensure <$3/video target is met. **Steps**:

1. Analyze cost breakdown from first videos
2. Implement caching for repeated API calls
3. Switch to GPT-3.5 for non-critical sections
4. Batch API requests where possible **Duration**: 3 hours **Dependencies**: Cost tracking from first videos **Deliverable**: Cost optimization strategy with 30% reduction **Priority**: P0

Task 3: Multi-Agent System Foundation

Description: Begin implementation of specialized AI agents for different tasks. **Steps**:

1. Design agent communication protocol
2. Implement TrendAnalyzer agent for content selection
3. Create QualityGuardian agent for content validation
4. Test agent coordination on sample workflow **Duration**: 4 hours **Dependencies**: Base AI pipeline working **Deliverable**: Two operational AI agents with coordination **Priority**: P1

Technical Teams (Under CTO)

Role: Backend Team Lead

Task 1: Channel Management API Completion

Description: Finalize all channel CRUD operations with business logic. **Steps**:

1. Implement channel limits (5 per user) validation
2. Add channel scheduling and automation settings
3. Create channel analytics endpoints
4. Implement soft delete for channel deactivation **Duration**: 4 hours **Dependencies**: Week 0 channel scaffold **Deliverable**: Complete channel management API with tests **Priority**: P0

Task 2: Video Generation Pipeline API

Description: Build API endpoints for video generation workflow. **Steps**:

1. Create POST /videos/generate endpoint with validation
2. Implement GET /videos/{id}/status for progress tracking
3. Add webhook callbacks for pipeline stages
4. Create batch generation endpoint for multiple videos **Duration**: 4 hours **Dependencies**: Queue system from Week 0 **Deliverable**: Video generation API with async processing **Priority**: P0

Task 3: Cost Tracking Integration

Description: Integrate real-time cost tracking into all API operations. **Steps**:

1. Add cost calculation to each API call
2. Implement cost aggregation by user/channel/video
3. Create cost alert system for threshold breaches
4. Add cost projection endpoints **Duration**: 3 hours **Dependencies**: Cost framework from Week 0 **Deliverable**: Integrated cost tracking with alerts **Priority**: P1

Role: API Developer Engineer

Task 1: YouTube Upload Implementation

Description: Complete YouTube video upload functionality with metadata. **Steps**:

1. Implement resumable upload for large video files
2. Add metadata (title, description, tags) management
3. Create thumbnail upload functionality
4. Implement upload status tracking **Duration**: 4 hours **Dependencies**: YouTube API client from Week 0 **Deliverable**: Working YouTube upload with 95% success rate **Priority**: P0

Task 2: Webhook Event System

Description: Build webhook infrastructure for external service callbacks. **Steps**:

1. Create webhook receiver endpoints
2. Implement webhook signature verification
3. Add event processing queue
4. Create retry mechanism for failed webhooks **Duration**: 3 hours **Dependencies**: Queue system **Deliverable**: Robust webhook handling system **Priority**: P1

Task 3: API Rate Limiting

Description: Implement rate limiting to prevent abuse and manage costs. **Steps**:

1. Add Redis-based rate limiting middleware
2. Configure limits per endpoint and user tier
3. Implement rate limit headers in responses
4. Create bypass for internal services **Duration**: 2 hours **Dependencies**: Redis setup **Deliverable**: Rate limiting with configurable thresholds **Priority**: P2

Role: Data Pipeline Engineer

Task 1: Video Processing Pipeline

Description: Build complete video generation pipeline from script to upload. **Steps**:

1. Implement script generation task with GPT integration
2. Create voice synthesis task with TTS services
3. Build video assembly task with FFmpeg
4. Add thumbnail generation with DALL-E/Stable Diffusion **Duration**: 4 hours **Dependencies**: AI model integrations **Deliverable**: End-to-end video processing pipeline **Priority**: P0

Task 2: Pipeline Monitoring Dashboard

Description: Create real-time monitoring for pipeline health and performance. **Steps**:

1. Add timing metrics for each pipeline stage
2. Implement success/failure tracking
3. Create queue depth monitoring
4. Add cost tracking per stage **Duration**: 3 hours **Dependencies**: Monitoring infrastructure **Deliverable**: Pipeline dashboard in Grafana **Priority**: P1

Task 3: Error Recovery Mechanisms

Description: Implement robust error handling and recovery for pipeline failures. **Steps**:

1. Add automatic retry logic with exponential backoff
2. Implement dead letter queue for failed tasks
3. Create manual retry interface
4. Add failure notification system **Duration**: 3 hours **Dependencies**: Pipeline implementation **Deliverable**: Self-healing pipeline with 90% recovery rate **Priority**: P1

Role: Integration Specialist

Task 1: OpenAI Integration Optimization

Description: Optimize GPT integration for quality and cost efficiency. **Steps**:

1. Implement response caching for common prompts
2. Add streaming responses for better UX
3. Create fallback from GPT-4 to GPT-3.5
4. Implement token counting and optimization **Duration**: 3 hours **Dependencies**: OpenAI API access **Deliverable**: Optimized OpenAI client with 40% cost reduction **Priority**: P0

Task 2: Stock Media API Integration

Description: Connect to stock media services for video assets. **Steps**:

1. Integrate Pexels API for video clips
2. Add Unsplash API for images
3. Implement asset caching and CDN storage
4. Create asset selection algorithm **Duration**: 4 hours **Dependencies**: API credentials **Deliverable**: Multi-source media library integration **Priority**: P1

Task 3: Payment System Foundation

Description: Begin Stripe integration for subscription management. **Steps**:

1. Set up Stripe customer creation
2. Implement subscription plans ($97, $297, $997)
3. Add webhook handlers for payment events
4. Create billing portal integration **Duration**: 3 hours **Dependencies**: Stripe account setup **Deliverable**: Basic payment flow with subscription support **Priority**: P2

Role: Frontend Team Lead

Task 1: Dashboard MVP Implementation

Description: Build functional dashboard showing key metrics and controls. **Steps**:

1. Create metrics cards showing channels, videos, costs
2. Implement channel switcher component
3. Add video generation trigger button
4. Display real-time cost tracking **Duration**: 4 hours **Dependencies**: API endpoints from backend **Deliverable**: Working dashboard with live data **Priority**: P0

Task 2: API Integration Layer

Description: Complete API client integration with error handling. **Steps**:

1. Implement all authentication flows
2. Add global error handling with user feedback
3. Create loading states for all async operations
4. Implement optimistic updates for better UX **Duration**: 3 hours **Dependencies**: Backend APIs **Deliverable**: Robust API layer with great UX **Priority**: P0

Task 3: Component Library Extension

Description: Build additional components needed for Week 2 features. **Steps**:

1. Create DataTable component for channel list

2. Build ProgressBar for video generation tracking
3. Add Modal system for confirmations
4. Create form components with validation **Duration**: 3 hours **Dependencies**: Design system **Deliverable**: 10 additional reusable components **Priority**: P1

Role: React Engineer

Task 1: Channel Management Interface

Description: Build complete channel management UI with CRUD operations. **Steps**:

1. Create channel list view with status indicators
2. Implement add/edit channel modal
3. Add channel settings panel
4. Build channel deletion with confirmation **Duration**: 4 hours **Dependencies**: Channel API endpoints **Deliverable**: Full channel management interface **Priority**: P0

Task 2: Video Generation Flow

Description: Create UI for triggering and monitoring video generation. **Steps**:

1. Build video generation form with options
2. Create progress tracking component
3. Implement cost estimation display
4. Add generation history list **Duration**: 4 hours **Dependencies**: Video generation API **Deliverable**: Complete video generation workflow UI **Priority**: P0

Task 3: Real-time Updates Implementation

Description: Add WebSocket connections for live updates. **Steps**:

1. Implement WebSocket connection manager
2. Add real-time video status updates
3. Create notification system for events
4. Implement connection status indicator **Duration**: 3 hours **Dependencies**: WebSocket endpoints **Deliverable**: Real-time updates for critical events **Priority**: P1

Role: Dashboard Specialist

Task 1: Analytics Dashboard Components

Description: Build data visualization components for metrics display. **Steps**:

1. Create revenue trend line chart

2. Build cost breakdown pie chart
3. Implement video performance bar chart
4. Add channel comparison charts **Duration**: 4 hours **Dependencies**: Recharts setup, API data **Deliverable**: 4 interactive chart components **Priority**: P1

Task 2: Performance Metrics Display

Description: Create real-time performance monitoring displays. **Steps**:

1. Build API latency monitor
2. Create video generation time tracker
3. Add success rate indicators
4. Implement cost per video gauge **Duration**: 3 hours **Dependencies**: Metrics endpoints **Deliverable**: Performance monitoring dashboard section **Priority**: P1

Task 3: Data Export Functionality

Description: Add ability to export analytics data for external analysis. **Steps**:

1. Implement CSV export for all data tables
2. Add date range selection for exports
3. Create PDF report generation
4. Add scheduled report functionality **Duration**: 2 hours **Dependencies**: Analytics data access **Deliverable**: Export functionality with multiple formats **Priority**: P2

Role: UI/UX Designer

Task 1: High-Fidelity Screen Designs

Description: Complete pixel-perfect designs for all Week 2 features. **Steps**:

1. Design video generation workflow screens
2. Create channel analytics dashboard layouts
3. Design settings and configuration pages
4. Add micro-interactions and transitions **Duration**: 4 hours **Dependencies**: Week 1 feedback **Deliverable**: 15 high-fidelity screen designs **Priority**: P0

Task 2: User Testing Session

Description: Conduct usability testing on Week 1 implementations. **Steps**:

1. Recruit 5 internal testers
2. Create testing script and tasks

3. Conduct recorded testing sessions
4. Compile findings and recommendations **Duration**: 4 hours **Dependencies**: Working dashboard **Deliverable**: Usability testing report with action items **Priority**: P1

Task 3: Design System Expansion

Description: Add new components to design system based on Week 1 needs. **Steps**:

1. Design data visualization components
2. Create empty states and error states
3. Add loading skeletons
4. Document component usage guidelines **Duration**: 3 hours **Dependencies**: Week 1 implementation feedback **Deliverable**: Expanded design system with 10 new components **Priority**: P2

Role: Platform Ops Lead

Task 1: Production Environment Preparation

Description: Set up production environment for beta launch preparation. **Steps**:

1. Configure production Docker swarm
2. Set up SSL certificates and domain
3. Implement security hardening
4. Create deployment runbooks **Duration**: 4 hours **Dependencies**: Week 0 infrastructure **Deliverable**: Production-ready environment **Priority**: P0

Task 2: Backup and Disaster Recovery Testing

Description: Validate backup and recovery procedures with real data. **Steps**:

1. Perform full system backup
2. Simulate failure scenario
3. Execute recovery procedure
4. Document recovery time (target: <4 hours) **Duration**: 3 hours **Dependencies**: Backup scripts from Week 0 **Deliverable**: Validated DR plan with RTO confirmed **Priority**: P1

Task 3: Performance Optimization

Description: Optimize infrastructure based on Week 1 load patterns. **Steps**:

1. Analyze resource utilization metrics
2. Tune Docker container limits
3. Optimize database connection pooling

4. Implement caching strategies **Duration**: 3 hours **Dependencies**: Week 1 monitoring data **Deliverable**: 30% performance improvement **Priority**: P1

Role: DevOps Engineer

Task 1: CI/CD Pipeline Enhancement

Description: Extend CI/CD pipeline with automated testing and deployment. **Steps**:

1. Add integration test stage to pipeline
2. Implement automated rollback on failure
3. Create staging deployment automation
4. Add security scanning stage **Duration**: 4 hours **Dependencies**: Week 0 CI/CD foundation **Deliverable**: Full CI/CD pipeline with 15-minute deployments **Priority**: P0

Task 2: Log Aggregation System

Description: Implement centralized logging for all services. **Steps**:

1. Set up ELK stack (Elasticsearch, Logstash, Kibana)
2. Configure log shipping from all containers
3. Create log parsing rules
4. Build debugging dashboards **Duration**: 4 hours **Dependencies**: All services running **Deliverable**: Centralized logging with search capabilities **Priority**: P1

Task 3: Auto-scaling Configuration

Description: Implement auto-scaling for critical services. **Steps**:

1. Define scaling metrics and thresholds
2. Configure horizontal pod autoscaling
3. Test scaling under load
4. Document scaling behaviors **Duration**: 3 hours **Dependencies**: Container orchestration **Deliverable**: Auto-scaling for API and workers **Priority**: P2

Role: Security Engineer

Task 1: Security Audit Implementation

Description: Conduct comprehensive security audit of Week 1 code. **Steps**:

1. Run OWASP dependency check
2. Perform static code analysis
3. Conduct authentication penetration testing

4. Document vulnerabilities and fixes **Duration**: 4 hours **Dependencies**: Week 1 code complete **Deliverable**: Security audit report with remediation plan **Priority**: P1

Task 2: API Security Hardening

Description: Implement additional API security measures. **Steps**:

1. Add API key management system
2. Implement request signing
3. Add DDoS protection rules
4. Create API access audit logging **Duration**: 3 hours **Dependencies**: API endpoints complete **Deliverable**: Hardened API with security layers **Priority**: P1

Task 3: Compliance Documentation

Description: Document compliance with data protection regulations. **Steps**:

1. Create data flow diagrams
2. Document PII handling procedures
3. Implement data retention policies
4. Create user data export functionality **Duration**: 3 hours **Dependencies**: Data pipeline complete **Deliverable**: Compliance documentation package **Priority**: P2

Role: QA Engineer

Task 1: End-to-End Test Suite

Description: Create comprehensive E2E tests for critical user flows. **Steps**:

1. Write tests for user registration and login
2. Create channel creation and management tests
3. Implement video generation flow tests
4. Add payment flow test scenarios **Duration**: 4 hours **Dependencies**: Features implemented **Deliverable**: 20 E2E tests with 90% pass rate **Priority**: P0

Task 2: Performance Testing

Description: Conduct load testing to validate system capacity. **Steps**:

1. Create load test scenarios with k6
2. Simulate 50 concurrent users
3. Test video generation under load

4. Document performance bottlenecks **Duration**: 3 hours **Dependencies**: System operational **Deliverable**: Performance test report with recommendations **Priority**: P1

Task 3: Bug Triage Process

Description: Establish bug management and prioritization process. **Steps**:

1. Set up bug tracking in JIRA
2. Define severity levels and SLAs
3. Create bug triage meeting schedule
4. Document escalation procedures **Duration**: 2 hours **Dependencies**: Testing in progress **Deliverable**: Bug management process documentation **Priority**: P2

AI Teams (Under VP of AI)

Role: AI/ML Team Lead

Task 1: Model Performance Optimization

Description: Optimize model inference for speed and cost based on Week 1 data. **Steps**:

1. Profile model inference times
2. Implement model quantization
3. Add response caching layer
4. Optimize batch processing **Duration**: 4 hours **Dependencies**: Week 1 inference data **Deliverable**: 50% faster inference with 30% cost reduction **Priority**: P0

Task 2: Quality Scoring System

Description: Implement automated quality scoring for generated content. **Steps**:

1. Define quality metrics (coherence, relevance, engagement)
2. Train quality prediction model
3. Integrate scoring into pipeline
4. Create quality threshold alerts **Duration**: 4 hours **Dependencies**: First videos generated **Deliverable**: Automated quality scoring with 85% accuracy **Priority**: P1

Task 3: A/B Testing Framework

Description: Build framework for testing different AI strategies. **Steps**:

1. Design experiment tracking system
2. Implement random assignment logic
3. Create metrics collection

4. Build results analysis dashboard **Duration**: 3 hours **Dependencies**: Multiple videos generated **Deliverable**: A/B testing system with first experiment **Priority**: P2

Role: ML Engineer

Task 1: Trend Prediction Deployment

Description: Deploy trend prediction model to production environment. **Steps**:

1. Containerize Prophet model
2. Set up model serving endpoint
3. Implement prediction caching
4. Add performance monitoring **Duration**: 4 hours **Dependencies**: Model training complete **Deliverable**: Production trend prediction with <500ms latency **Priority**: P0

Task 2: Feature Engineering Pipeline

Description: Build automated feature extraction for model inputs. **Steps**:

1. Create YouTube metrics feature extractor
2. Implement trending signals processor
3. Build feature store with versioning
4. Add feature monitoring **Duration**: 3 hours **Dependencies**: Data pipeline **Deliverable**: Automated feature pipeline with 50+ features **Priority**: P1

Task 3: Model Retraining Automation

Description: Set up automated model retraining based on performance. **Steps**:

1. Define retraining triggers
2. Implement training pipeline
3. Add model validation stage
4. Create automatic deployment **Duration**: 3 hours **Dependencies**: ML infrastructure **Deliverable**: Automated retraining with validation **Priority**: P2

Role: Data Team Lead

Task 1: Data Quality Framework

Description: Implement data quality monitoring and validation. **Steps**:

1. Define data quality metrics
2. Create validation rules
3. Implement anomaly detection

4. Build quality dashboard **Duration**: 4 hours **Dependencies**: Data pipeline operational **Deliverable**: Data quality monitoring with 99% accuracy **Priority**: P1

Task 2: Analytics Pipeline Scaling

Description: Optimize analytics pipeline for increased load. **Steps**:

1. Implement data partitioning
2. Add query optimization
3. Create materialized views
4. Set up incremental processing **Duration**: 3 hours **Dependencies**: Week 1 data volume **Deliverable**: 10x faster analytics queries **Priority**: P1

Task 3: Reporting API Development

Description: Build comprehensive reporting API for dashboards. **Steps**:

1. Design API schema
2. Implement aggregation endpoints
3. Add caching layer
4. Create documentation **Duration**: 3 hours **Dependencies**: Analytics database **Deliverable**: Reporting API with 10 endpoints **Priority**: P2

Role: Data Engineer

Task 1: Real-time Data Streaming

Description: Implement real-time data streaming for live metrics. **Steps**:

1. Set up Kafka for event streaming
2. Create producers for all services
3. Implement stream processing
4. Build real-time aggregations **Duration**: 4 hours **Dependencies**: Services generating events **Deliverable**: Real-time data pipeline with <1s latency **Priority**: P1

Task 2: YouTube Analytics Integration

Description: Build comprehensive YouTube analytics data collection. **Steps**:

1. Implement YouTube Analytics API v2 client
2. Create scheduled collection jobs
3. Build data transformation pipeline

4. Add historical data backfill **Duration**: 4 hours **Dependencies**: YouTube API access **Deliverable**: Complete YouTube metrics collection **Priority**: P0

Task 3: Cost Data Pipeline

Description: Build detailed cost tracking and attribution system. **Steps**:

1. Integrate API usage tracking
2. Create cost allocation logic
3. Build cost aggregation pipeline
4. Implement cost alerts **Duration**: 3 hours **Dependencies**: API integrations **Deliverable**: Real-time cost tracking per video **Priority**: P1

Role: Analytics Engineer

Task 1: KPI Dashboard Development

Description: Build comprehensive KPI tracking and visualization. **Steps**:

1. Define 20 core KPIs
2. Create calculation logic
3. Build real-time dashboards
4. Add drill-down capabilities **Duration**: 4 hours **Dependencies**: Data pipeline **Deliverable**: KPI dashboard with 20 metrics **Priority**: P1

Task 2: Cohort Analysis Implementation

Description: Build cohort analysis for user behavior tracking. **Steps**:

1. Define cohort segments
2. Implement retention calculations
3. Create cohort visualizations
4. Add predictive analytics **Duration**: 3 hours **Dependencies**: User data available **Deliverable**: Cohort analysis with retention metrics **Priority**: P2

Task 3: Revenue Attribution Model

Description: Create model for attributing revenue to platform features. **Steps**:

1. Design attribution logic
2. Implement tracking system
3. Create attribution reports

4. Add ROI calculations **Duration**: 3 hours **Dependencies**: Revenue data **Deliverable**: Revenue attribution system **Priority**: P2

Week 1 Critical Milestones

Day 1-2: Foundation & Integration

- ✅ All Week 0 work verified and integrated
- ✅ First API endpoints live and tested
- ✅ Basic UI connected to backend
- ✅ AI models accessible via API

Day 3: First Video Attempt

- ✅ **CRITICAL**: First automated video generated end-to-end
- ✅ Cost tracking verified (<$3)
- ✅ Quality score calculated
- ✅ Video uploaded to YouTube successfully

Day 4: Optimization & Scaling

- ✅ 10 test videos generated
- ✅ Performance bottlenecks identified
- ✅ Cost optimizations implemented
- ✅ Quality improvements applied

Day 5: Integration & Planning

- ✅ All teams synchronized
- ✅ Week 2 plan finalized
- ✅ Beta user feedback incorporated
- ✅ Technical debt logged

Success Metrics for Week 1

Technical Metrics

1. **Video Generation**: 10+ videos successfully generated
2. **Cost Per Video**: <$3 achieved and tracked
3. **Pipeline Success Rate**: >80% without manual intervention
4. **API Uptime**: >99% for all services
5. **Response Time**: <500ms for all API endpoints

Quality Metrics

1. **Content Quality Score**: >70/100 average
2. **YouTube Compliance**: Zero policy violations
3. **Generation Time**: <10 minutes per video
4. **Error Rate**: <5% for all operations

Team Metrics

1. **Story Points Completed**: 80% of planned work
2. **Integration Tests Passing**: >90%
3. **Code Coverage**: >70% for new code
4. **Documentation**: All APIs documented

Risk Mitigation for Week 1

High-Risk Areas

1. **YouTube API Quotas**: Monitor usage, implement caching aggressively
2. **Cost Overruns**: Real-time monitoring, automatic fallbacks to cheaper models
3. **Quality Issues**: Manual review for first 20 videos, adjustment period
4. **Integration Failures**: 4 PM daily integration testing sessions

Contingency Plans

1. **If video generation fails**: Manual pipeline execution with debugging
2. **If costs exceed $5/video**: Immediate switch to all economy models
3. **If YouTube rejects uploads**: Manual upload with policy review
4. **If performance degrades**: Scale back features, focus on core flow

Week 1 Deliverables Summary

Must-Have Deliverables

1. ✅ 10 test videos generated and uploaded
2. ✅ Cost tracking operational and verified
3. ✅ Basic dashboard with real-time metrics
4. ✅ Channel management fully functional
5. ✅ Quality scoring system active

Nice-to-Have Deliverables

1. ⭕ Payment system foundation
2. ⭕ Advanced analytics dashboards
3. ⭕ Automated retraining pipeline
4. ⭕ Complete design system

Handoff to Week 2

Technical Handoffs

1. **Backend → Frontend**: Complete API documentation, WebSocket events defined
2. **AI → Backend**: Optimized models deployed, cost projections validated
3. **Platform Ops → All**: Monitoring dashboards configured, CI/CD fully automated
4. **Frontend → Product**: User feedback compiled, UI pain points identified

Process Handoffs

1. **QA → Development**: Bug backlog prioritized
2. **Security → Platform Ops**: Vulnerabilities patched
3. **Data → Analytics**: KPIs defined and tracking
4. **Product → CEO**: Beta user feedback synthesized

Week 2 Preparation

1. **Feature Focus**: Channel automation and scheduling
2. **Scale Target**: 50 videos/day capacity
3. **Quality Target**: 80/100 average score
4. **Cost Target**: $2.50/video
5. **User Target**: First 3 beta users onboarded

Daily Standup Topics

Monday (Day 1)

- Week 0 completion verification
- Integration point confirmation
- Blockers identification

Tuesday (Day 2)

- API endpoint testing results
- UI-Backend connection status
- First video generation prep

Wednesday (Day 3)

- **CRITICAL**: First video generation
- Cost tracking verification
- Quality assessment

Thursday (Day 4)

- Video generation at scale (10 videos)
- Performance optimization findings
- Cost reduction implementation

Friday (Day 5)

- Week 1 retrospective
- Week 2 planning
- Beta user feedback review
- Technical debt assessment

Communication Protocols

Escalation Path

1. **Technical Blockers**: Team Lead → CTO → CEO
2. **Cost Overruns**: VP AI → CFO → CEO
3. **Quality Issues**: Product Owner → VP AI → CTO
4. **Security Concerns**: Security Engineer → CTO → CEO

Meeting Schedule

- **9:00 AM**: Daily standup (all teams)
- **11:00 AM**: Technical sync (leads only)
- **2:00 PM**: Product review (as needed)
- **4:00 PM**: Integration testing (all teams)
- **5:00 PM**: End-of-day status (leads)

Documentation Requirements

- All code must have inline documentation
- API changes require updated OpenAPI specs
- Architecture decisions logged in ADR format
- Daily progress updates in Confluence

Definition of Done for Week 1

A task is considered DONE when:

1. ✅ Code is written and reviewed
2. ✅ Tests are written and passing (>70% coverage)
3. ✅ Documentation is updated
4. ✅ Integration tests pass
5. ✅ Performance benchmarks met
6. ✅ Security review completed
7. ✅ Deployed to staging environment
8. ✅ Product Owner acceptance received