YTEMpire Week 1 Execution Plan

Executive Summary

Week 1 focuses on building core functionality: authentication, channel management, basic video generation pipeline, and initial AI integrations. Target: Generate first 10 automated videos with <$3 cost tracking.

Sprint Goals

- **Primary**: End-to-end video generation pipeline functional
- **Secondary**: Multi-channel management system operational
- **Tertiary**: Cost tracking and optimization framework active
- **Stretch**: First beta user onboarded for testing

Leadership Team

Role: CEO

Task 1: Beta User Recruitment Kickoff [P1]

Description: Launch beta user recruitment campaign targeting 10 early adopters. **Steps**:

1. Finalize beta user criteria (YouTube creators with 3+ channels)
2. Create landing page with application form
3. Reach out to 50 potential beta users via LinkedIn/Twitter
4. Schedule initial interviews with top 20 candidates **Duration**: 3 hours **Dependencies**: Product messaging from Product Owner **Deliverable**: 20+ qualified beta user applications

Task 2: Investor Update Preparation [P2]

Description: Prepare Week 1 progress update for stakeholders showing MVP momentum. **Steps**:

1. Compile development metrics from all teams
2. Create demo video of current functionality
3. Update financial projections based on cost data
4. Draft investor update email with key achievements **Duration**: 2 hours **Dependencies**: Demo from CTO, cost data from VP of AI **Deliverable**: Investor update package with demo video

Task 3: Strategic Partnership Outreach [P2]

Description: Initiate conversations with potential strategic partners for content and distribution. **Steps**:

1. Identify top 10 strategic partners (content libraries, API providers)

2. Draft partnership proposal templates
3. Send initial outreach emails
4. Schedule exploratory calls for Week 2 **Duration**: 3 hours **Dependencies**: Product roadmap from Product Owner **Deliverable**: Partnership pipeline with 5+ active conversations

Role: CTO (Technical Director)

Task 1: System Integration Testing [P0]

Description: Conduct comprehensive integration testing across all services. **Steps**:

1. Define integration test scenarios (20+ critical paths)
2. Execute tests between Frontend ↔ Backend ↔ AI services
3. Document integration issues and assign fixes
4. Verify data flow from input to YouTube upload **Duration**: 4 hours **Dependencies**: All services running from Week 0 **Deliverable**: Integration test report with 90%+ pass rate

Task 2: Performance Baseline Establishment [P1]

Description: Measure and document system performance baselines for all critical operations. **Steps**:

1. Load test API endpoints (target: 100 req/sec)
2. Measure video generation pipeline speed (target: <10 min)
3. Profile database query performance
4. Document bottlenecks and optimization opportunities **Duration**: 3 hours **Dependencies**: Monitoring stack from Platform Ops **Deliverable**: Performance baseline report with optimization roadmap

Task 3: Architecture Review Session [P1]

Description: Conduct architecture review with all technical leads to refine design. **Steps**:

1. Review Week 0 implementation against architecture
2. Identify architectural debt and shortcuts taken
3. Prioritize architectural improvements for Week 2
4. Update architecture documentation **Duration**: 2 hours **Dependencies**: Week 0 code complete **Deliverable**: Updated architecture with improvement backlog

Task 4: Security Audit Phase 1 [P2]

Description: Perform initial security audit of authentication and data handling. **Steps**:

1. Review authentication implementation for vulnerabilities
2. Audit API endpoint security and rate limiting

3. Check secrets management implementation
4. Create security improvement checklist **Duration**: 3 hours **Dependencies**: Security framework from Security Engineer **Deliverable**: Security audit report with critical fixes identified

Role: VP of AI

Task 1: AI Cost Optimization Implementation [P0]

Description: Implement intelligent cost optimization across all AI services. **Steps**:

1. Create tiered prompt templates (simple → complex)
2. Implement caching layer for repeated AI calls
3. Set up fallback chains (GPT-4 → GPT-3.5 → Claude)
4. Test cost reduction strategies (achieve 30% reduction) **Duration**: 4 hours **Dependencies**: Cost tracking from Week 0 **Deliverable**: Optimized AI pipeline with <$1.50/video AI costs

Task 2: Quality Scoring System Deployment [P1]

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

1. Implement multi-factor quality scoring (0-100 scale)
2. Create quality thresholds for auto-approval (>80) vs review (<60)
3. Build feedback loop for quality improvement
4. Test with 20+ generated videos **Duration**: 3 hours **Dependencies**: ML models from ML Engineer **Deliverable**: Quality scoring system with 85% accuracy

Task 3: Multi-Model Orchestration [P1]

Description: Implement orchestration layer for multiple AI models. **Steps**:

1. Design model routing logic based on content type
2. Implement parallel model execution where possible
3. Create model performance monitoring
4. Test with different content scenarios **Duration**: 4 hours **Dependencies**: Model serving infrastructure from ML Team Lead **Deliverable**: Orchestration layer handling 5+ models

Task 4: Trend Intelligence System Alpha [P2]

Description: Deploy alpha version of trend detection and prediction system. **Steps**:

1. Connect to 10+ data sources (YouTube, Google Trends, Reddit)
2. Implement trend scoring algorithm
3. Create trend visualization dashboard

4. Test predictions against historical data **Duration**: 3 hours **Dependencies**: Data pipeline from Data Engineer **Deliverable**: Trend system with 70% accuracy on backtesting

Role: Product Owner

Task 1: User Acceptance Criteria Definition [P0]

Description: Define detailed acceptance criteria for Week 1 features. **Steps**:

1. Write acceptance criteria for channel management features
2. Define video generation workflow requirements
3. Specify dashboard metric requirements
4. Create UAT test scenarios **Duration**: 3 hours **Dependencies**: User stories from Week 0 **Deliverable**: Acceptance criteria document with 50+ test cases

Task 2: Beta User Onboarding Flow Design [P1]

Description: Design and document complete beta user onboarding experience. **Steps**:

1. Map onboarding journey from signup to first video
2. Create onboarding checklist and tooltips
3. Design help documentation and FAQs
4. Build feedback collection mechanisms **Duration**: 4 hours **Dependencies**: UI designs from UI/UX Designer **Deliverable**: Onboarding flow with <30 min time to first video

Task 3: Feature Prioritization for Week 2 [P2]

Description: Prioritize feature backlog based on Week 1 learnings. **Steps**:

1. Analyze Week 1 development velocity
2. Gather feedback from development teams
3. Re-prioritize backlog using value/effort matrix
4. Create Week 2 sprint plan **Duration**: 2 hours **Dependencies**: Team velocity data from all leads **Deliverable**: Prioritized Week 2 backlog with story points

Technical Team (Under CTO)

Role: Backend Team Lead

Task 1: Channel Management API Development [P0]

Description: Build complete CRUD API for YouTube channel management. **Steps**:

1. Implement channel creation with YouTube OAuth
2. Build channel listing with pagination and filtering

3. Create channel analytics aggregation endpoints
4. Add channel-specific settings management **Duration**: 4 hours **Dependencies**: YouTube API integration from API Developer **Deliverable**: Channel management API with 15+ endpoints

Task 2: Video Queue System Implementation [P0]

Description: Build robust video generation queue with priority management. **Steps**:

1. Implement priority queue using Redis
2. Create queue monitoring endpoints
3. Build retry mechanism for failed jobs
4. Add queue analytics and reporting **Duration**: 4 hours **Dependencies**: Redis setup from Week 0 **Deliverable**: Queue system handling 100+ concurrent jobs

Task 3: Batch Processing Framework [P1]

Description: Implement batch processing for multiple video generation. **Steps**:

1. Create batch job submission endpoint
2. Implement parallel processing logic
3. Build batch status tracking
4. Add batch cancellation and modification **Duration**: 3 hours **Dependencies**: Queue system complete **Deliverable**: Batch processing handling 50+ videos

Task 4: API Rate Limiting Enhancement [P2]

Description: Implement sophisticated rate limiting per user and endpoint. **Steps**:

1. Create tiered rate limits by user subscription
2. Implement sliding window rate limiting
3. Add rate limit headers to responses
4. Build rate limit analytics **Duration**: 2 hours **Dependencies**: Authentication system from Week 0 **Deliverable**: Rate limiting system with per-user quotas

Role: API Developer Engineer

Task 1: YouTube Upload Automation [P0]

Description: Implement automated video upload to YouTube with metadata. **Steps**:

1. Build video upload endpoint with resumable uploads
2. Implement metadata (title, description, tags) management
3. Create thumbnail upload functionality

4. Add scheduling and visibility controls **Duration**: 4 hours **Dependencies**: YouTube OAuth from Week 0 **Deliverable**: Upload system with 95% success rate

Task 2: Analytics Data Synchronization [P1]

Description: Build system to sync YouTube Analytics data. **Steps**:

1. Implement YouTube Analytics API integration
2. Create scheduled sync jobs (every 6 hours)
3. Build incremental update logic
4. Add data validation and error handling **Duration**: 3 hours **Dependencies**: Database schema from Backend Lead **Deliverable**: Analytics sync with <1 hour data freshness

Task 3: Webhook Event System [P1]

Description: Implement webhook system for real-time event notifications. **Steps**:

1. Create webhook subscription management
2. Implement event publishing system
3. Build retry logic for failed deliveries
4. Add webhook logs and debugging tools **Duration**: 3 hours **Dependencies**: Message queue from Backend Lead **Deliverable**: Webhook system with 99% delivery rate

Task 4: Content Moderation API [P2]

Description: Build content moderation endpoints for generated content. **Steps**:

1. Integrate YouTube's content ID check
2. Implement profanity and sensitive content filters
3. Create manual review queue endpoints
4. Add moderation analytics **Duration**: 2 hours **Dependencies**: AI quality scoring from VP of AI **Deliverable**: Moderation API catching 95% of policy violations

Role: Data Pipeline Engineer

Task 1: Real-time Analytics Pipeline [P0]

Description: Build real-time analytics processing for video performance. **Steps**:

1. Implement stream processing for view events
2. Create real-time aggregations (5-min windows)
3. Build materialized views for dashboards

4. Add anomaly detection for metrics **Duration**: 4 hours **Dependencies**: Event streaming from Week 0 **Deliverable**: Real-time pipeline with <1 min latency

Task 2: Cost Aggregation Pipeline [P1]

Description: Build pipeline to aggregate costs across all services. **Steps**:

1. Create cost collection from all API calls
2. Implement cost allocation by video/channel
3. Build cost forecasting based on usage
4. Add cost alerts and thresholds **Duration**: 3 hours **Dependencies**: AI cost tracking from VP of AI **Deliverable**: Cost pipeline accurate to $0.01

Task 3: Data Export System [P2]

Description: Implement data export functionality for users. **Steps**:

1. Create CSV/JSON export endpoints
2. Implement scheduled report generation
3. Build data anonymization for exports
4. Add export job management **Duration**: 3 hours **Dependencies**: Analytics pipeline complete **Deliverable**: Export system handling 10GB+ datasets

Role: Integration Specialist

Task 1: n8n Video Generation Workflow [P0]

Description: Create complete video generation workflow in n8n. **Steps**:

1. Build trigger node for video requests
2. Implement AI content generation nodes
3. Create video rendering workflow
4. Add YouTube upload node **Duration**: 4 hours **Dependencies**: n8n setup from Week 0 **Deliverable**: End-to-end workflow generating videos in <10 min

Task 2: Third-party Media Integration [P1]

Description: Integrate stock media APIs for video content. **Steps**:

1. Integrate Pexels API for stock footage
2. Connect Pixabay for images
3. Add Freesound for audio effects

4. Implement media caching layer **Duration**: 3 hours **Dependencies**: API framework from Week 0 **Deliverable**: Media library with 10,000+ assets

Task 3: Payment System Integration [P2]

Description: Integrate Stripe for beta user payments. **Steps**:

1. Implement Stripe checkout flow
2. Create subscription management endpoints
3. Build usage-based billing logic
4. Add payment webhooks handling **Duration**: 3 hours **Dependencies**: User management from Backend Lead **Deliverable**: Payment system ready for beta

Role: Frontend Team Lead

Task 1: Channel Management UI [P0]

Description: Build complete channel management interface. **Steps**:

1. Create channel list view with cards
2. Implement add/edit channel modals
3. Build channel settings panel
4. Add channel switching dropdown **Duration**: 4 hours **Dependencies**: Channel API from Backend Lead **Deliverable**: Channel UI managing 5+ channels

Task 2: Video Generation Interface [P0]

Description: Create intuitive video generation workflow UI. **Steps**:

1. Build video topic input with suggestions
2. Create generation options panel
3. Implement progress tracking UI
4. Add preview and approval interface **Duration**: 4 hours **Dependencies**: Video generation API from Backend **Deliverable**: Video generation UI with <5 clicks to generate

Task 3: Real-time Updates Implementation [P1]

Description: Implement WebSocket connections for real-time updates. **Steps**:

1. Create WebSocket connection manager
2. Implement real-time notification system
3. Build live progress indicators

4. Add connection status indicators **Duration**: 3 hours **Dependencies**: WebSocket endpoints from Backend **Deliverable**: Real-time updates with <1 sec latency

Role: React Engineer

Task 1: Dashboard Components Development [P0]

Description: Build core dashboard components with real data. **Steps**:

1. Create metrics cards with trend indicators
2. Build channel performance table
3. Implement recent videos list
4. Add quick actions panel **Duration**: 4 hours **Dependencies**: Analytics API from Backend **Deliverable**: Dashboard showing real metrics

Task 2: Form Validation System [P1]

Description: Implement comprehensive form validation across application. **Steps**:

1. Create validation rules library
2. Implement real-time validation feedback
3. Build error message system
4. Add form state management **Duration**: 3 hours **Dependencies**: API error responses from Backend **Deliverable**: Form validation with <100ms feedback

Task 3: Responsive Design Implementation [P2]

Description: Ensure application works on different screen sizes. **Steps**:

1. Implement responsive grid system
2. Create mobile-friendly navigation
3. Optimize touch interactions
4. Test on 5+ device sizes **Duration**: 3 hours **Dependencies**: UI designs from UI/UX Designer **Deliverable**: Responsive UI working on 1280px+ screens

Role: Dashboard Specialist

Task 1: Analytics Dashboard Development [P0]

Description: Build comprehensive analytics dashboard with charts. **Steps**:

1. Implement revenue tracking chart
2. Create view trends visualization
3. Build engagement metrics heatmap

4. Add comparative channel analysis **Duration**: 4 hours **Dependencies**: Analytics data from Backend **Deliverable**: Dashboard with 8+ interactive charts

Task 2: Real-time Metrics Display [P1]

Description: Implement real-time metric updates without page refresh. **Steps**:

1. Create polling mechanism for metrics
2. Implement smooth update animations
3. Build metric change indicators
4. Add customizable refresh rates **Duration**: 3 hours **Dependencies**: Real-time API from Backend **Deliverable**: Metrics updating every 30 seconds

Task 3: Export and Reporting UI [P2]

Description: Build interface for data export and report generation. **Steps**:

1. Create export configuration modal
2. Build report template selector
3. Implement download progress indicator
4. Add export history view **Duration**: 2 hours **Dependencies**: Export API from Data Pipeline Engineer **Deliverable**: Export UI handling large datasets

Role: UI/UX Designer

Task 1: Video Generation Flow Refinement [P0]

Description: Refine video generation UX based on initial testing. **Steps**:

1. Conduct usability testing with 5 team members
2. Identify friction points in generation flow
3. Design improved workflow with fewer steps
4. Create interactive prototype for validation **Duration**: 4 hours **Dependencies**: Initial UI implementation from Frontend Lead **Deliverable**: Refined designs reducing clicks by 30%

Task 2: Mobile-Responsive Designs [P1]

Description: Create responsive design specifications for tablet/mobile. **Steps**:

1. Design responsive breakpoints for key screens
2. Create mobile navigation patterns
3. Optimize touch targets for mobile

4. Document responsive behavior rules **Duration**: 3 hours **Dependencies**: Desktop designs from Week 0 **Deliverable**: Responsive design system for 3 breakpoints

Task 3: Error State Designs [P2]

Description: Design comprehensive error states and empty states. **Steps**:

1. Create error message templates
2. Design empty state illustrations
3. Build loading state animations
4. Document error handling patterns **Duration**: 3 hours **Dependencies**: Error scenarios from QA Engineer **Deliverable**: Error state design library

Role: Platform Ops Lead

Task 1: Production Environment Setup [P0]

Description: Configure production environment for beta launch. **Steps**:

1. Provision production servers (local + cloud hybrid)
2. Configure production database with replication
3. Set up production monitoring and alerting
4. Implement backup and disaster recovery **Duration**: 4 hours **Dependencies**: Infrastructure from Week 0 **Deliverable**: Production environment with 99.9% uptime target

Task 2: Auto-scaling Configuration [P1]

Description: Implement auto-scaling for video processing workloads. **Steps**:

1. Configure CPU/memory-based scaling triggers
2. Implement queue-depth-based scaling
3. Set up scaling notifications
4. Test scaling under load **Duration**: 3 hours **Dependencies**: Kubernetes setup from DevOps **Deliverable**: Auto-scaling handling 10x load spikes

Task 3: Disaster Recovery Testing [P2]

Description: Test disaster recovery procedures end-to-end. **Steps**:

1. Simulate database failure and recovery
2. Test backup restoration procedures
3. Verify data integrity after recovery

4. Document recovery time objectives (RTO) **Duration**: 3 hours **Dependencies**: Backup systems from Week 0 **Deliverable**: DR test report with <4 hour RTO

Role: DevOps Engineer

Task 1: CI/CD Pipeline Enhancement [P0]

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

1. Add automated unit test execution
2. Implement integration test stage
3. Create automated staging deployment
4. Add rollback mechanisms **Duration**: 4 hours **Dependencies**: Test suites from QA Engineer **Deliverable**: CI/CD with 15-min deployment cycle

Task 2: Container Optimization [P1]

Description: Optimize Docker containers for size and performance. **Steps**:

1. Implement multi-stage builds
2. Minimize image sizes (<100MB targets)
3. Add container health checks
4. Optimize layer caching **Duration**: 3 hours **Dependencies**: Docker setup from Week 0 **Deliverable**: Container sizes reduced by 50%

Task 3: Log Aggregation System [P2]

Description: Implement centralized log aggregation and search. **Steps**:

1. Deploy ELK stack or similar
2. Configure log shipping from all services
3. Create log parsing rules
4. Build log search dashboard **Duration**: 3 hours **Dependencies**: Services running from Week 0 **Deliverable**: Centralized logs with <5 sec search

Role: Security Engineer

Task 1: API Security Hardening [P0]

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

1. Add API key rotation mechanism
2. Implement request signing
3. Add IP whitelisting capabilities

4. Enable audit logging for all API calls **Duration**: 4 hours **Dependencies**: API development from Backend team **Deliverable**: Hardened API with security score >90

Task 2: Data Encryption Implementation [P1]

Description: Implement encryption for sensitive data at rest and in transit. **Steps**:

1. Enable TLS 1.3 for all connections
2. Implement database field encryption
3. Add encrypted file storage
4. Create key rotation procedures **Duration**: 3 hours **Dependencies**: Database setup from Backend Lead **Deliverable**: All sensitive data encrypted

Task 3: Security Monitoring Setup [P2]

Description: Deploy security monitoring and intrusion detection. **Steps**:

1. Configure fail2ban for brute force protection
2. Set up anomaly detection rules
3. Implement security event logging
4. Create security dashboard **Duration**: 3 hours **Dependencies**: Monitoring stack from Platform Ops **Deliverable**: Security monitoring catching 95% of threats

Role: QA Engineer

Task 1: End-to-End Test Suite Development [P0]

Description: Build comprehensive E2E test suite for critical user flows. **Steps**:

1. Write tests for complete video generation flow
2. Create channel management test scenarios
3. Implement dashboard verification tests
4. Add cross-browser test execution **Duration**: 4 hours **Dependencies**: Frontend from React team **Deliverable**: E2E suite with 30+ test scenarios

Task 2: API Testing Framework [P1]

Description: Implement automated API testing framework. **Steps**:

1. Create Postman/Insomnia collections
2. Write contract tests for all endpoints
3. Implement load testing scenarios

4. Add API performance benchmarks **Duration**: 3 hours **Dependencies**: API documentation from Backend **Deliverable**: API test suite with 95% coverage

Task 3: Performance Testing Implementation [P2]

Description: Create performance testing suite for system optimization. **Steps**:

1. Build load testing scenarios with k6
2. Create stress testing configurations
3. Implement performance regression tests
4. Generate performance reports **Duration**: 3 hours **Dependencies**: Production environment from Platform Ops **Deliverable**: Performance tests validating SLAs

AI Team (Under VP of AI)

Role: AI/ML Team Lead

Task 1: Model Deployment Pipeline [P0]

Description: Create automated model deployment pipeline with versioning. **Steps**:

1. Implement model registry with MLflow
2. Create A/B testing framework for models
3. Build automated model validation
4. Add rollback capabilities **Duration**: 4 hours **Dependencies**: Infrastructure from Platform Ops **Deliverable**: Model deployment with <10 min rollout

Task 2: Feature Engineering Pipeline [P1]

Description: Build automated feature engineering for ML models. **Steps**:

1. Create feature extraction from video metadata
2. Implement trend signal features
3. Build engagement prediction features
4. Add feature versioning **Duration**: 3 hours **Dependencies**: Data pipeline from Data Engineer **Deliverable**: Feature pipeline generating 50+ features

Task 3: Model Performance Monitoring [P2]

Description: Implement comprehensive model performance tracking. **Steps**:

1. Create prediction logging system
2. Build accuracy tracking dashboards
3. Implement drift detection

4. Add automated retraining triggers **Duration**: 3 hours **Dependencies**: Monitoring stack from Platform Ops **Deliverable**: Model monitoring dashboard with alerts

Role: ML Engineer

Task 1: Content Generation Model Enhancement [P0]

Description: Improve content generation model for better quality. **Steps**:

1. Fine-tune prompts for different content types
2. Implement style consistency mechanisms
3. Add fact-checking validation
4. Test with 50+ video generations **Duration**: 4 hours **Dependencies**: GPT-4 access from VP of AI **Deliverable**: Model with 85% quality score average

Task 2: Thumbnail Generation System [P1]

Description: Build AI-powered thumbnail generation system. **Steps**:

1. Integrate Stable Diffusion for image generation
2. Create thumbnail template system
3. Implement A/B testing framework
4. Add CTR prediction model **Duration**: 4 hours **Dependencies**: Image generation APIs from VP of AI **Deliverable**: Thumbnail system with 8% CTR average

Task 3: Voice Synthesis Optimization [P2]

Description: Optimize voice synthesis for quality and cost. **Steps**:

1. Compare ElevenLabs vs Google TTS quality
2. Implement voice caching system
3. Create voice style templates
4. Add pronunciation correction **Duration**: 3 hours **Dependencies**: TTS APIs from VP of AI **Deliverable**: Voice system with 90% naturalness score

Role: Data Engineer (AI Team)

Task 1: Training Data Management System [P0]

Description: Build system for managing ML training datasets. **Steps**:

1. Create data versioning system
2. Implement data validation pipeline
3. Build training data statistics dashboard

4. Add data lineage tracking **Duration**: 4 hours **Dependencies**: Database from Backend team **Deliverable**: Training data system with full versioning

Task 2: Feature Store Implementation [P1]

Description: Deploy feature store for real-time ML serving. **Steps**:

1. Set up Feast or similar feature store
2. Implement online/offline feature sync
3. Create feature monitoring
4. Add feature access APIs **Duration**: 3 hours **Dependencies**: Redis from Backend team **Deliverable**: Feature store with <10ms serving latency

Task 3: ML Data Pipeline Automation [P2]

Description: Automate data collection and preparation for ML models. **Steps**:

1. Create automated data collection jobs
2. Implement data cleaning pipelines
3. Build data augmentation system
4. Add pipeline monitoring **Duration**: 3 hours **Dependencies**: ETL framework from Week 0 **Deliverable**: Automated pipeline processing 1M+ records/day

Role: Data Engineer 2 (AI Team)

Task 1: Real-time Inference Pipeline [P0]

Description: Build real-time inference pipeline for production models. **Steps**:

1. Implement model serving with TorchServe
2. Create request batching for efficiency
3. Add result caching layer
4. Build fallback mechanisms **Duration**: 4 hours **Dependencies**: Models from ML Engineer **Deliverable**: Inference pipeline with <100ms p95 latency

Task 2: Analytics Data Lake Setup [P1]

Description: Create data lake for analytics and ML training. **Steps**:

1. Configure S3-compatible storage
2. Implement data partitioning strategy
3. Create data catalog with Hive metastore

4. Add data governance policies **Duration**: 3 hours **Dependencies**: Storage infrastructure from Platform Ops **Deliverable**: Data lake with 10TB+ capacity

Task 3: Streaming Analytics Implementation [P2]

Description: Build streaming analytics for real-time insights. **Steps**:

1. Deploy Apache Flink for stream processing
2. Create real-time aggregations
3. Build streaming dashboards
4. Add alerting for anomalies **Duration**: 3 hours **Dependencies**: Event streaming from Week 0 **Deliverable**: Streaming analytics with <1 sec latency

Role: Analytics Engineer

Task 1: Business Metrics Dashboard [P0]

Description: Build comprehensive business metrics dashboard. **Steps**:

1. Create revenue tracking visualizations
2. Build user engagement metrics
3. Implement cost analysis views
4. Add ROI calculations **Duration**: 4 hours **Dependencies**: Data from Backend APIs **Deliverable**: Executive dashboard with 15+ KPIs

Task 2: Channel Performance Analytics [P1]

Description: Develop detailed channel performance analytics. **Steps**:

1. Create channel comparison metrics
2. Build growth trend analysis
3. Implement competitor benchmarking
4. Add performance predictions **Duration**: 3 hours **Dependencies**: YouTube data from API Developer **Deliverable**: Channel analytics with predictive insights

Task 3: A/B Testing Analytics Framework [P2]

Description: Build framework for analyzing A/B test results. **Steps**:

1. Create statistical significance calculators
2. Build test result visualizations
3. Implement automated test analysis

4. Add recommendation engine **Duration**: 3 hours **Dependencies**: A/B testing data from ML Team Lead **Deliverable**: A/B testing dashboard with statistical analysis

Daily Schedule - Week 1

Monday (Day 6)

- 9:00 AM: Sprint 1 Planning (2 hours)
- 11:00 AM: Team breakout sessions
- 2:00 PM: Development time
- 4:30 PM: End-of-day sync

Tuesday (Day 7)

- 9:00 AM: Daily standup (15 min)
- 9:30 AM: Focus time (no meetings)
- 2:00 PM: API integration testing
- 4:00 PM: Progress check

Wednesday (Day 8)

- 9:00 AM: Daily standup (15 min)
- 10:00 AM: Mid-sprint review
- 2:00 PM: Cross-team debugging session
- 4:00 PM: Cost optimization review

Thursday (Day 9)

- 9:00 AM: Daily standup (15 min)
- 10:00 AM: Security review
- 2:00 PM: Performance testing
- 4:00 PM: Beta user prep meeting

Friday (Day 10)

- 9:00 AM: Daily standup (15 min)
- 10:00 AM: Sprint demo preparation
- 2:00 PM: Sprint 1 Demo (all hands)
- 3:30 PM: Sprint retrospective
- 4:30 PM: Week 2 planning

Success Metrics - Week 1

Must Achieve (P0 Items)

□ 10+ videos generated end-to-end
□ 5 YouTube channels connected and managed
□ Cost tracking showing <$3 per video
□ 95% API success rate
□ Frontend dashboard displaying real data

Should Achieve (P1 Items)

□ Real-time updates working
□ Quality scoring on all videos
□ Analytics pipeline processing
□ A/B testing framework operational
□ First beta user successfully onboarded

Nice to Have (P2 Items)

□ Mobile responsive design complete
□ Advanced security monitoring active
□ Streaming analytics operational
□ Payment system integrated
□ 20+ beta user applications

Risk Mitigation - Week 1

Technical Risks

- **Integration Complexity**: Daily integration testing sessions
- **Performance Issues**: Continuous profiling and optimization
- **AI Cost Overruns**: Hourly cost monitoring and alerts
- **Data Quality**: Automated validation at every stage

Process Risks

- **Scope Creep**: Strict P0/P1/P2 enforcement
- **Team Dependencies**: Twice-daily dependency checks
- **Knowledge Silos**: Pair programming mandatory
- **Testing Delays**: Parallel test development

Deliverables Checklist - End of Week 1

Working Software

□ Multi-channel management system

- □ Video generation pipeline (10 min end-to-end)
- □ Cost tracking system (<$3/video verified)
- □ Real-time analytics dashboard
- □ Quality scoring system

Documentation

- □ API documentation (50+ endpoints)
- □ User guide for beta testers
- □ System architecture (updated)
- □ Deployment procedures
- □ Security audit report

Metrics

- □ 50+ test videos generated
- □ 5+ channels fully configured
- □ <$150 total AI costs
- □ 95% test coverage on critical paths
- □ <10 min video generation time

Demo Script - Friday 2:00 PM

Part 1: User Journey (15 min)

1. User registration and onboarding
2. Connect YouTube channel via OAuth
3. Generate first video from trending topic
4. Review and approve generated content
5. Automated upload to YouTube
6. View analytics dashboard

Part 2: Technical Deep Dive (10 min)

1. Show cost breakdown per video
2. Demonstrate quality scoring
3. Display real-time metrics
4. Show A/B testing framework
5. Review system performance metrics

Part 3: Beta Preview (5 min)

1. Show beta user applications

2. Demo improved UX from feedback
3. Preview Week 2 features
4. Q&A session

Transition to Week 2

Handoff Items

- Updated backlog with velocity data
- Technical debt documentation
- Performance optimization opportunities
- Beta user feedback summary
- Architecture refinements needed

Week 2 Preview

- Scale to 50+ videos/day
- Onboard 3-5 beta users
- Implement advanced AI features
- Optimize for <$2 per video
- Launch closed beta program

This document represents the complete Week 1 execution plan for YTEMpire MVP. Building on Week 0's foundation, Week 1 delivers core functionality enabling automated video generation at scale with comprehensive cost tracking and quality assurance.