YTEmpire Week 1 Execution Plan

Week 1 Overview

Sprint Goal: Achieve first end-to-end video generation with basic pipeline operational
Target Metrics: 1 test video generated, 5 API integrations complete, 10 core components built
Key Milestone: Proof of concept demonstrating 95% automation feasibility

Executive Leadership

Role: CEO/Founder

Task 1: Beta User Interview Sessions

Description: Conduct detailed interviews with 3 potential beta users to validate MVP features. **Steps**:

1. Schedule 1-hour sessions with qualified prospects
2. Conduct discovery interviews using prepared questionnaire
3. Document pain points and feature priorities
4. Share insights with Product Owner and team leads **Duration**: 6 hours (3x2 hours) **Dependencies**: Beta user recruitment from Week 0 **Deliverable**: User interview synthesis document **Priority**: P1 **Status Checkpoint**: Tuesday EOD

Task 2: Investor Update Preparation

Description: Prepare Week 1 progress update for advisors and potential investors. **Steps**:

1. Compile technical progress metrics from all teams
2. Create visual progress dashboard
3. Document key wins and challenges
4. Schedule update calls for Week 2 **Duration**: 3 hours **Dependencies**: Progress reports from team leads **Deliverable**: Investor update deck **Priority**: P2 **Status Checkpoint**: Friday 2 PM

Task 3: Partnership Outreach

Description: Initiate conversations with potential content and technology partners. **Steps**:

1. Reach out to 5 YouTube growth agencies
2. Contact stock media API providers for partnerships
3. Explore affiliate network opportunities
4. Document partnership terms and opportunities **Duration**: 4 hours **Dependencies**: None **Deliverable**: Partnership pipeline document **Priority**: P2 **Status Checkpoint**: Thursday EOD

Role: Product Owner

Task 1: User Story Sprint Planning

Description: Break down Week 1 features into detailed user stories with acceptance criteria. **Steps**:

1. Write 15-20 user stories for Week 1 features
2. Define acceptance criteria for each story
3. Assign story points with team leads
4. Load stories into project management tool **Duration**: 4 hours **Dependencies**: Feature priority matrix from Week 0 **Deliverable**: Sprint backlog with sized stories **Priority**: P0 **Status Checkpoint**: Monday 11 AM

Task 2: MVP Feature Specification v1.0

Description: Create detailed specifications for core MVP features based on Week 0 learnings. **Steps**:

1. Document channel management feature specs
2. Define video generation workflow requirements
3. Specify dashboard metrics and visualizations
4. Review specs with engineering leads **Duration**: 6 hours **Dependencies**: User interview insights from CEO **Deliverable**: Feature specification document v1.0 **Priority**: P0 **Status Checkpoint**: Wednesday EOD

Task 3: Quality Criteria Definition

Description: Establish quality benchmarks for automated content generation. **Steps**:

1. Define minimum acceptable video quality score (0-10 scale)
2. Set performance benchmarks (generation time, cost)
3. Create content policy compliance checklist
4. Document quality gate criteria **Duration**: 3 hours **Dependencies**: Input from AI Team Lead **Deliverable**: Quality standards document **Priority**: P1 **Status Checkpoint**: Thursday 3 PM

Technical Leadership

Role: CTO/Technical Director

Task 1: Architecture Review & Refinement

Description: Refine technical architecture based on Week 0 discoveries and team feedback. **Steps**:

1. Conduct architecture review session with team leads
2. Update architecture diagrams with implementation details

3. Identify and resolve architectural risks
4. Create architecture decision records (ADRs) **Duration**: 4 hours **Dependencies**: Week 0 architecture document **Deliverable**: Architecture v1.1 with ADRs **Priority**: P0 **Status Checkpoint**: Monday 4 PM

Task 2: Integration Points Coordination

Description: Coordinate API contracts and integration points between teams. **Steps**:

1. Facilitate API contract definition session
2. Review and approve OpenAPI specifications
3. Establish integration testing strategy
4. Create integration timeline with milestones **Duration**: 4 hours **Dependencies**: API designs from team leads **Deliverable**: Integration specification document **Priority**: P0 **Status Checkpoint**: Tuesday 3 PM

Task 3: Performance Benchmarking Setup

Description: Establish performance benchmarking framework and baseline metrics. **Steps**:

1. Define key performance indicators (latency, throughput)
2. Set up performance testing environment
3. Create automated performance test suite
4. Run initial baseline benchmarks **Duration**: 4 hours **Dependencies**: Monitoring stack from DevOps **Deliverable**: Performance benchmark report v1 **Priority**: P1 **Status Checkpoint**: Friday 11 AM

Task 4: Technical Risk Assessment

Description: Identify and plan mitigation for top technical risks. **Steps**:

1. Conduct risk assessment workshop with leads
2. Prioritize risks by impact and probability
3. Create mitigation plans for top 5 risks
4. Assign risk owners and tracking **Duration**: 3 hours **Dependencies**: Architecture review complete **Deliverable**: Technical risk register with mitigations **Priority**: P1 **Status Checkpoint**: Wednesday 2 PM

Role: VP of AI

Task 1: End-to-End ML Pipeline Implementation

Description: Build and test complete ML pipeline from trend detection to content generation. **Steps**:

1. Integrate trend detection with content generation
2. Implement pipeline orchestration logic

3. Add monitoring and logging at each stage
4. Test with 5 different content scenarios **Duration**: 8 hours **Dependencies**: ML components from AI team **Deliverable**: Working ML pipeline v1 **Priority**: P0 **Status Checkpoint**: Wednesday EOD

Task 2: Cost Optimization Implementation

Description: Implement cost tracking and optimization strategies for AI operations. **Steps**:

1. Add cost tracking to each API call
2. Implement caching layer for repeated requests
3. Set up fallback model chain (GPT-4 → GPT-3.5)
4. Create cost dashboard and alerts **Duration**: 6 hours **Dependencies**: Metrics collection from Analytics Engineer **Deliverable**: Cost optimization system v1 **Priority**: P0 **Status Checkpoint**: Thursday EOD

Task 3: Model Quality Assurance Framework

Description: Establish quality testing framework for AI-generated content. **Steps**:

1. Create quality scoring algorithm
2. Build test dataset with labeled examples
3. Implement automated quality gates
4. Set up A/B testing infrastructure **Duration**: 5 hours **Dependencies**: Quality criteria from Product Owner **Deliverable**: Quality assurance framework **Priority**: P1 **Status Checkpoint**: Friday 3 PM

Backend Team

Role: Backend Team Lead

Task 1: Core API Implementation Sprint

Description: Implement core CRUD APIs for users, channels, and videos. **Steps**:

1. Implement user management endpoints (register, login, profile)
2. Create channel CRUD operations with validation
3. Build video management APIs
4. Add comprehensive error handling **Duration**: 8 hours **Dependencies**: Database migrations from Week 0 **Deliverable**: 15+ working API endpoints **Priority**: P0 **Status Checkpoint**: Tuesday EOD

Task 2: Authentication & Authorization System

Description: Complete JWT-based auth system with role-based access control. **Steps**:

1. Implement JWT refresh token mechanism
2. Add role-based permissions (admin, user)

3. Create middleware for route protection
4. Implement rate limiting per user **Duration**: 6 hours **Dependencies**: Authentication scaffolding from Week 0 **Deliverable**: Complete auth system **Priority**: P0 **Status Checkpoint**: Monday EOD

Task 3: API Documentation & Testing

Description: Create comprehensive API documentation and test coverage. **Steps**:

1. Generate OpenAPI documentation
2. Write unit tests for all endpoints (target 80% coverage)
3. Create integration test suite
4. Set up Postman collection for team **Duration**: 4 hours **Dependencies**: Core APIs complete **Deliverable**: API docs and test suite **Priority**: P1 **Status Checkpoint**: Thursday 4 PM

Role: API Developer Engineer

Task 1: Channel Management Service

Description: Build complete channel management service with YouTube integration. **Steps**:

1. Implement channel creation with YouTube OAuth
2. Add channel settings and customization
3. Build channel analytics endpoints
4. Create channel-video relationship management **Duration**: 8 hours **Dependencies**: YouTube API plan from Integration Specialist **Deliverable**: Channel management service **Priority**: P0 **Status Checkpoint**: Wednesday EOD

Task 2: Video Queue Management

Description: Implement video generation queue with status tracking. **Steps**:

1. Create video job submission endpoint
2. Implement queue status and position tracking
3. Add job cancellation and retry logic
4. Build real-time status updates via WebSocket **Duration**: 6 hours **Dependencies**: Message queue from Data Pipeline Engineer **Deliverable**: Video queue management system **Priority**: P0 **Status Checkpoint**: Thursday EOD

Task 3: Webhook Handlers

Description: Implement webhook handlers for external service callbacks. **Steps**:

1. Create YouTube upload completion handler
2. Implement payment webhook handlers

3. Add video processing status callbacks
4. Set up webhook security and verification **Duration**: 4 hours **Dependencies**: External API integrations **Deliverable**: Webhook handling system **Priority**: P1 **Status Checkpoint**: Friday 2 PM

Role: Data Pipeline Engineer

Task 1: Video Processing Pipeline

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

1. Implement script → audio conversion flow
2. Create audio + visuals → video assembly
3. Add thumbnail generation step
4. Integrate YouTube upload with retry logic **Duration**: 10 hours **Dependencies**: AI pipeline from ML team **Deliverable**: End-to-end video pipeline **Priority**: P0 **Status Checkpoint**: Thursday EOD

Task 2: Parallel Processing Implementation

Description: Enable parallel processing for multiple video generation. **Steps**:

1. Implement Celery worker pool configuration
2. Add task routing based on resource requirements
3. Create GPU/CPU task separation
4. Test with 5 concurrent video generations **Duration**: 6 hours **Dependencies**: Video processing pipeline **Deliverable**: Parallel processing capability **Priority**: P1 **Status Checkpoint**: Friday 4 PM

Role: Integration Specialist

Task 1: YouTube API Integration

Description: Complete YouTube Data API v3 integration with quota management. **Steps**:

1. Implement OAuth2 flow for 15 accounts
2. Create video upload with metadata
3. Add quota tracking and rotation logic
4. Implement channel statistics fetching **Duration**: 8 hours **Dependencies**: API credentials from Platform Ops **Deliverable**: Complete YouTube integration **Priority**: P0 **Status Checkpoint**: Tuesday EOD

Task 2: OpenAI API Integration

Description: Integrate OpenAI API for script generation with error handling. **Steps**:

1. Implement GPT-4/GPT-3.5 API wrapper

2. Add retry logic with exponential backoff
3. Create prompt template management
4. Implement response caching **Duration**: 6 hours **Dependencies**: API keys from VP of AI **Deliverable**: OpenAI integration module **Priority**: P0 **Status Checkpoint**: Wednesday 3 PM

Task 3: Google TTS Integration

Description: Integrate Google Text-to-Speech for voice synthesis. **Steps**:

1. Set up Google Cloud TTS client
2. Implement voice selection logic
3. Add SSML support for enhanced speech
4. Create audio file management **Duration**: 4 hours **Dependencies**: Google Cloud credentials **Deliverable**: Google TTS integration **Priority**: P0 **Status Checkpoint**: Thursday 11 AM

Frontend Team

Role: Frontend Team Lead

Task 1: Dashboard Shell Implementation

Description: Build main dashboard shell with routing and state management. **Steps**:

1. Implement main layout with responsive design
2. Set up React Router for navigation
3. Configure Zustand stores for state
4. Add loading and error boundaries **Duration**: 6 hours **Dependencies**: Component library from Week 0 **Deliverable**: Working dashboard shell **Priority**: P0 **Status Checkpoint**: Tuesday 3 PM

Task 2: API Integration Layer

Description: Create API service layer with error handling and caching. **Steps**:

1. Set up Axios with interceptors
2. Implement API service classes
3. Add request/response caching
4. Create error handling utilities **Duration**: 5 hours **Dependencies**: API documentation from Backend **Deliverable**: API integration layer **Priority**: P0 **Status Checkpoint**: Wednesday 4 PM

Task 3: Real-time Updates Implementation

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

1. Set up WebSocket client

2. Implement reconnection logic
3. Create event handlers for status updates
4. Integrate with Zustand stores **Duration**: 4 hours **Dependencies**: WebSocket endpoints from Backend **Deliverable**: Real-time update system **Priority**: P1 **Status Checkpoint**: Friday 11 AM

Role: React Engineer

Task 1: Authentication Flow UI

Description: Complete authentication flow with all screens and logic. **Steps**:

1. Finish login/register forms with validation
2. Implement protected route wrapper
3. Add token refresh logic
4. Create user profile management UI **Duration**: 6 hours **Dependencies**: Auth endpoints from Backend **Deliverable**: Complete auth flow **Priority**: P0 **Status Checkpoint**: Tuesday EOD

Task 2: Channel Management Interface

Description: Build channel creation and management interface. **Steps**:

1. Create channel creation wizard
2. Build channel list with cards
3. Add channel settings panel
4. Implement channel switching logic **Duration**: 8 hours **Dependencies**: Channel APIs from Backend **Deliverable**: Channel management UI **Priority**: P0 **Status Checkpoint**: Thursday EOD

Task 3: Video Queue Interface

Description: Create video queue visualization with status tracking. **Steps**:

1. Build queue list component
2. Add progress indicators for each video
3. Create video detail modal
4. Implement queue actions (pause, cancel) **Duration**: 6 hours **Dependencies**: Video queue APIs **Deliverable**: Video queue interface **Priority**: P1 **Status Checkpoint**: Friday 3 PM

Role: Dashboard Specialist

Task 1: Metrics Dashboard Components

Description: Build dashboard components for key metrics display. **Steps**:

1. Create metrics cards (videos, revenue, costs)

2. Implement trend indicators
3. Add period comparison (day/week/month)
4. Create loading and error states **Duration**: 6 hours **Dependencies**: Dashboard data from Analytics **Deliverable**: Metrics components **Priority**: P1 **Status Checkpoint**: Wednesday EOD

Task 2: Cost Tracking Visualization

Description: Build cost tracking dashboard with breakdown charts. **Steps**:

1. Create cost per video display
2. Build cost breakdown pie chart
3. Add cost trend line chart
4. Implement budget alerts UI **Duration**: 5 hours **Dependencies**: Cost data endpoints **Deliverable**: Cost tracking dashboard **Priority**: P1 **Status Checkpoint**: Thursday 4 PM

Task 3: Channel Performance Charts

Description: Implement channel performance visualization charts. **Steps**:

1. Create views/subscribers line chart
2. Build engagement rate metrics
3. Add revenue tracking chart
4. Implement channel comparison view **Duration**: 6 hours **Dependencies**: Recharts setup, Analytics API **Deliverable**: Performance charts **Priority**: P2 **Status Checkpoint**: Friday EOD

Role: UI/UX Designer

Task 1: High-Fidelity Dashboard Designs

Description: Create pixel-perfect designs for main dashboard views. **Steps**:

1. Design dashboard overview screen
2. Create channel management layouts
3. Design video queue interface
4. Add micro-interactions and animations **Duration**: 8 hours **Dependencies**: Wireframes from Week 0 **Deliverable**: Figma designs for 5 key screens **Priority**: P0 **Status Checkpoint**: Tuesday 2 PM

Task 2: Component Library Expansion

Description: Design additional components for the design system. **Steps**:

1. Design data visualization components
2. Create form components and validation states

3. Design modal and overlay patterns
4. Document component usage guidelines **Duration**: 6 hours **Dependencies**: Base design system **Deliverable**: Expanded component library **Priority**: P1 **Status Checkpoint**: Thursday 2 PM

Task 3: Mobile Responsive Designs

Description: Create responsive designs for tablet and mobile views. **Steps**:

1. Adapt dashboard for tablet (768px)
2. Design mobile navigation pattern
3. Create responsive data tables
4. Document responsive breakpoints **Duration**: 4 hours **Dependencies**: Desktop designs complete **Deliverable**: Responsive design specifications **Priority**: P2 **Status Checkpoint**: Friday 4 PM

Platform Operations Team

Role: Platform Ops Lead

Task 1: Production Environment Setup

Description: Configure production environment on local server. **Steps**:

1. Complete server OS installation and hardening
2. Configure production Docker environment
3. Set up production databases
4. Implement backup automation **Duration**: 8 hours **Dependencies**: Hardware delivery confirmation **Deliverable**: Production environment ready **Priority**: P0 **Status Checkpoint**: Tuesday EOD

Task 2: Disaster Recovery Testing

Description: Test disaster recovery procedures and document results. **Steps**:

1. Simulate system failure scenarios
2. Test backup restoration process
3. Verify data integrity after recovery
4. Document recovery time objectives (RTO) **Duration**: 4 hours **Dependencies**: Backup system from Week 0 **Deliverable**: DR test report **Priority**: P1 **Status Checkpoint**: Thursday 3 PM

Task 3: Capacity Planning

Description: Plan resource allocation for expected load. **Steps**:

1. Calculate resource needs for 50 users
2. Plan scaling triggers and thresholds

3. Configure resource monitoring alerts
4. Document capacity expansion plan **Duration**: 3 hours **Dependencies**: Performance benchmarks **Deliverable**: Capacity planning document **Priority**: P1 **Status Checkpoint**: Friday 2 PM

Role: DevOps Engineer

Task 1: CI/CD Pipeline Implementation

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

1. Configure GitHub Actions for all repositories
2. Set up automated testing on PR
3. Implement staging deployment pipeline
4. Add production deployment with approvals **Duration**: 8 hours **Dependencies**: GitHub repos from Week 0 **Deliverable**: Full CI/CD pipeline **Priority**: P0 **Status Checkpoint**: Wednesday EOD

Task 2: Container Orchestration

Description: Finalize Docker Compose orchestration for all services. **Steps**:

1. Complete Docker Compose for 10+ services
2. Configure service dependencies
3. Implement health checks
4. Test rolling updates **Duration**: 6 hours **Dependencies**: Service Dockerfiles **Deliverable**: Complete container orchestration **Priority**: P0 **Status Checkpoint**: Tuesday 4 PM

Task 3: Log Aggregation Setup

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

1. Configure log shipping from containers
2. Set up log parsing and indexing
3. Create log search interface
4. Implement log retention policies **Duration**: 4 hours **Dependencies**: Monitoring stack **Deliverable**: Centralized logging system **Priority**: P1 **Status Checkpoint**: Friday 11 AM

Role: Security Engineer

Task 1: Security Audit & Hardening

Description: Conduct security audit and implement hardening measures. **Steps**:

1. Run vulnerability scans on all services
2. Implement security headers

3. Configure Web Application Firewall rules
4. Set up intrusion detection **Duration**: 6 hours **Dependencies**: Services deployed **Deliverable**: Security audit report with fixes **Priority**: P0 **Status Checkpoint**: Wednesday 3 PM

Task 2: API Security Implementation

Description: Implement API security measures including rate limiting. **Steps**:

1. Configure API rate limiting by tier
2. Implement API key management
3. Set up request validation
4. Add security monitoring **Duration**: 5 hours **Dependencies**: API endpoints from Backend **Deliverable**: Secured API layer **Priority**: P0 **Status Checkpoint**: Thursday EOD

Task 3: Compliance Checklist

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

1. Review GDPR compliance requirements
2. Implement data encryption at rest
3. Set up audit logging
4. Create privacy policy draft **Duration**: 4 hours **Dependencies**: Data flow documentation **Deliverable**: Compliance checklist and fixes **Priority**: P1 **Status Checkpoint**: Friday 3 PM

Role: QA Engineer

Task 1: End-to-End Test Suite

Description: Create automated end-to-end test suite for critical paths. **Steps**:

1. Write E2E tests for user registration/login
2. Create channel creation test flow
3. Test video generation pipeline
4. Implement dashboard verification tests **Duration**: 8 hours **Dependencies**: All features implemented **Deliverable**: E2E test suite (10+ scenarios) **Priority**: P1 **Status Checkpoint**: Thursday EOD

Task 2: Performance Testing

Description: Conduct performance testing and establish baselines. **Steps**:

1. Create load test scenarios
2. Test API endpoints under load
3. Measure response times and throughput

4. Document performance baselines **Duration**: 6 hours **Dependencies**: Test environment ready **Deliverable**: Performance test report **Priority**: P1 **Status Checkpoint**: Friday EOD

Task 3: Test Data Management

Description: Set up test data generation and management system. **Steps**:

1. Create test data generators
2. Build data seeding scripts
3. Implement test data cleanup
4. Document test data scenarios **Duration**: 4 hours **Dependencies**: Database schema **Deliverable**: Test data management system **Priority**: P2 **Status Checkpoint**: Wednesday 4 PM

AI Team

Role: AI/ML Team Lead

Task 1: Trend Detection Model Integration

Description: Integrate and optimize trend detection model for production. **Steps**:

1. Deploy trend detection model to serving infrastructure
2. Implement real-time data pipeline
3. Add prediction caching layer
4. Test with live YouTube data **Duration**: 8 hours **Dependencies**: Model serving infrastructure **Deliverable**: Production trend detection system **Priority**: P0 **Status Checkpoint**: Wednesday EOD

Task 2: Content Quality Scoring

Description: Implement ML-based content quality scoring system. **Steps**:

1. Train quality prediction model
2. Create scoring API endpoint
3. Implement feedback loop for improvement
4. Set up A/B testing framework **Duration**: 6 hours **Dependencies**: Training data from Data Engineer **Deliverable**: Quality scoring system **Priority**: P1 **Status Checkpoint**: Thursday 4 PM

Task 3: Model Monitoring Dashboard

Description: Set up model performance monitoring and alerting. **Steps**:

1. Implement model metric collection
2. Create performance dashboards
3. Set up drift detection

4. Configure performance alerts **Duration**: 4 hours **Dependencies**: Monitoring infrastructure **Deliverable**: Model monitoring system **Priority**: P1 **Status Checkpoint**: Friday 2 PM

Role: ML Engineer

Task 1: Script Generation Pipeline

Description: Build production-ready script generation using GPT models. **Steps**:

1. Implement prompt engineering framework
2. Create content personalization layer
3. Add quality validation checks
4. Test with 20 different topics **Duration**: 8 hours **Dependencies**: OpenAI integration **Deliverable**: Script generation service **Priority**: P0 **Status Checkpoint**: Tuesday EOD

Task 2: Voice Synthesis Integration

Description: Integrate multiple TTS services with fallback logic. **Steps**:

1. Integrate Google TTS as primary
2. Add ElevenLabs as premium option
3. Implement fallback chain
4. Create voice selection algorithm **Duration**: 6 hours **Dependencies**: TTS API credentials **Deliverable**: Voice synthesis service **Priority**: P0 **Status Checkpoint**: Wednesday 4 PM

Task 3: Thumbnail Generation

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

1. Integrate Stable Diffusion API
2. Create thumbnail prompt templates
3. Add text overlay generation
4. Implement A/B test variants **Duration**: 5 hours **Dependencies**: Image generation API **Deliverable**: Thumbnail generation service **Priority**: P1 **Status Checkpoint**: Friday 11 AM

Role: Data Team Lead

Task 1: Analytics Data Pipeline

Description: Build analytics pipeline for business metrics. **Steps**:

1. Set up event streaming infrastructure
2. Create data transformation jobs
3. Build aggregation pipelines

4. Implement data quality checks **Duration**: 8 hours **Dependencies**: Event collection from Analytics Engineer **Deliverable**: Analytics pipeline v1 **Priority**: P1 **Status Checkpoint**: Wednesday EOD

Task 2: Feature Store Implementation

Description: Set up feature store for ML model serving. **Steps**:

1. Design feature storage schema
2. Implement feature computation pipeline
3. Create feature serving API
4. Add feature versioning **Duration**: 6 hours **Dependencies**: ML pipeline architecture **Deliverable**: Feature store v1 **Priority**: P1 **Status Checkpoint**: Thursday 3 PM

Task 3: Data Quality Framework

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

1. Create data validation rules
2. Implement quality metrics collection
3. Set up data quality dashboards
4. Add alerting for data issues **Duration**: 4 hours **Dependencies**: Data pipeline **Deliverable**: Data quality framework **Priority**: P2 **Status Checkpoint**: Friday 4 PM

Role: Data Engineer

Task 1: Training Data Collection

Description: Build system for collecting and storing ML training data. **Steps**:

1. Create data collection endpoints
2. Implement data labeling interface
3. Set up training data storage
4. Build data versioning system **Duration**: 8 hours **Dependencies**: Data schema from Data Lead **Deliverable**: Training data system **Priority**: P1 **Status Checkpoint**: Tuesday EOD

Task 2: Real-time Streaming Setup

Description: Implement real-time data streaming for trend detection. **Steps**:

1. Set up Kafka/Redis Streams
2. Create streaming consumers
3. Implement stream processing

4. Add stream monitoring **Duration**: 6 hours **Dependencies**: Infrastructure from DevOps **Deliverable**: Streaming data pipeline **Priority**: P1 **Status Checkpoint**: Thursday 11 AM

Task 3: Batch Processing Jobs

Description: Create batch processing jobs for analytics and ML. **Steps**:

1. Implement daily aggregation jobs
2. Create ML training data preparation
3. Set up job scheduling
4. Add job monitoring **Duration**: 5 hours **Dependencies**: Data pipeline **Deliverable**: Batch processing system **Priority**: P2 **Status Checkpoint**: Friday 3 PM

Role: Analytics Engineer

Task 1: Metrics Collection System

Description: Implement comprehensive metrics collection across all services. **Steps**:

1. Instrument all services with metrics
2. Create metrics aggregation logic
3. Implement cost calculation
4. Set up metrics API **Duration**: 8 hours **Dependencies**: Service deployments **Deliverable**: Metrics collection system **Priority**: P0 **Status Checkpoint**: Tuesday 4 PM

Task 2: Business Dashboard Data

Description: Prepare data models for business dashboards. **Steps**:

1. Create revenue calculation queries
2. Build user analytics aggregations
3. Implement channel performance metrics
4. Optimize query performance **Duration**: 6 hours **Dependencies**: Analytics pipeline **Deliverable**: Dashboard data models **Priority**: P1 **Status Checkpoint**: Thursday EOD

Task 3: Cost Analytics Implementation

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

1. Track API costs per operation
2. Calculate infrastructure costs
3. Create cost attribution model

4. Build cost optimization recommendations **Duration**: 5 hours **Dependencies**: Metrics collection **Deliverable**: Cost analytics system **Priority**: P1 **Status Checkpoint**: Friday EOD

Daily Sprint Schedule

Monday (Sprint Planning Day)

- **9:00 AM**: Sprint planning meeting (2 hours)
- **11:00 AM**: Team breakout sessions
- **2:00 PM**: Technical architecture review
- **4:00 PM**: API contract finalization
- **5:00 PM**: Day 1 progress check

Tuesday (Core Development)

- **9:00 AM**: Daily standup
- **9:30 AM**: Focused development time
- **2:00 PM**: Backend-Frontend sync
- **3:00 PM**: AI team integration review
- **5:00 PM**: Progress checkpoint

Wednesday (Integration Focus)

- **9:00 AM**: Daily standup
- **10:00 AM**: Integration testing session
- **2:00 PM**: Cross-team debugging
- **4:00 PM**: Performance review
- **5:00 PM**: Mid-week demo prep

Thursday (Testing & Refinement)

- **9:00 AM**: Daily standup
- **10:00 AM**: QA testing session
- **2:00 PM**: Bug triage meeting
- **3:00 PM**: Security review
- **5:00 PM**: End-to-end test run

Friday (Demo & Planning)

- **9:00 AM**: Daily standup
- **10:00 AM**: Final integration test

- **2:00 PM**: Sprint demo (all hands)
- **4:00 PM**: Retrospective
- **5:00 PM**: Week 2 planning

Integration Milestones

By End of Day 2 (Tuesday)

- [ ] Authentication system operational
- [ ] Basic APIs responding
- [ ] Frontend connected to backend
- [ ] First AI model integrated

By End of Day 3 (Wednesday)

- [ ] YouTube API uploading videos
- [ ] Complete pipeline test run
- [ ] Dashboard showing real data
- [ ] Cost tracking operational

By End of Day 5 (Friday)

- [ ] First video generated end-to-end
- [ ] 10+ API endpoints tested
- [ ] 5+ dashboard screens functional
- [ ] All teams integrated

Success Metrics for Week 1

Critical Success Criteria (P0)

- [x] Generate 1 complete video through automated pipeline
- [x] YouTube upload successful
- [x] Cost tracking showing <$3 per video
- [x] Authentication and user management working
- [x] Core APIs operational (15+ endpoints)

Important Achievements (P1)

- [x] Dashboard displaying real metrics
- [x] 5 concurrent video generations tested
- [x] Quality scoring operational
- [x] CI/CD pipeline deploying to staging
- [x] Security baseline implemented

Stretch Goals (P2)

- [x] 10 videos generated successfully
- [x] Complete test coverage >70%
- [x] Performance benchmarks established
- [x] Mobile responsive design started

Risk Mitigation Tracker

Technical Risks

Risk	Probability	Impact	Mitigation	Owner	Status
API Rate Limits	High	High	Implement caching, rotation	Integration Specialist	In Progress
Video Quality Issues	Medium	High	Multiple model fallbacks	ML Engineer	Planned
Performance Bottlenecks	Medium	Medium	Profiling and optimization	DevOps Engineer	Monitoring
Integration Failures	Low	High	Comprehensive error handling	Backend Lead	In Progress

Operational Risks

Risk	Probability	Impact	Mitigation	Owner	Status
Team Dependencies	Medium	Medium	Daily syncs, clear interfaces	CTO	Active
Scope Creep	Medium	High	Strict sprint planning	Product Owner	Controlled
Hardware Issues	Low	Critical	Cloud backup ready	Platform Ops Lead	Prepared

Communication Plan

Scheduled Meetings

- **Daily Standup**: 9:00 AM (15 minutes) - Blockers and progress
- **Integration Sync**: 2:00 PM Tuesday/Thursday - Cross-team coordination
- **Leadership Check-in**: 5:00 PM Daily - Executive team sync
- **Sprint Demo**: 2:00 PM Friday - All hands demonstration

Escalation Protocol

1. **Level 1** (Team Lead): Technical blockers, resource needs
2. **Level 2** (CTO/VP AI): Architecture decisions, major delays
3. **Level 3** (CEO): Budget changes, scope modifications, external blockers

Communication Channels

- **#sprint-week1**: Main sprint coordination
- **#integration**: Cross-team integration issues
- **#blockers-urgent**: Critical blocking issues
- **#wins**: Celebrate completed milestones
- **#help-needed**: Request assistance

Testing Checklist

Unit Testing (By Wednesday)

- [ ] Backend: 80% coverage on core services
- [ ] Frontend: Component tests for all UI elements
- [ ] AI: Model inference tests passing

Integration Testing (By Thursday)

- [ ] API endpoints tested with Frontend
- [ ] ML pipeline integrated with Backend
- [ ] YouTube upload from pipeline working
- [ ] Database transactions verified

End-to-End Testing (By Friday)

- [ ] Complete user journey tested
- [ ] Video generation start to finish
- [ ] Cost calculation accurate
- [ ] Performance within targets

Deliverables Summary

Backend Team

- 15+ REST API endpoints operational
- Authentication system complete
- Video processing pipeline working
- YouTube integration uploading videos

Frontend Team

- Dashboard shell with navigation
- Authentication flow complete
- Channel management interface

- Real-time updates working

AI/ML Team

- Script generation producing content
- Voice synthesis operational
- Trend detection integrated
- Quality scoring implemented

Platform Ops

- Production environment ready
- CI/CD pipeline operational
- Monitoring and logging active
- Security measures implemented

Week 2 Preparation

Handoff Requirements

- [ ] All P0 tasks completed
- [ ] Integration tests passing
- [ ] Documentation updated
- [ ] Known issues logged
- [ ] Week 2 backlog prepared

Key Decisions Needed

- [ ] Feature priorities for Week 2
- [ ] Resource allocation adjustments
- [ ] Technology choices validation
- [ ] Architecture refinements

Lessons Learned Topics

- [ ] Integration challenges
- [ ] Performance bottlenecks
- [ ] Team coordination
- [ ] Technical debt identified

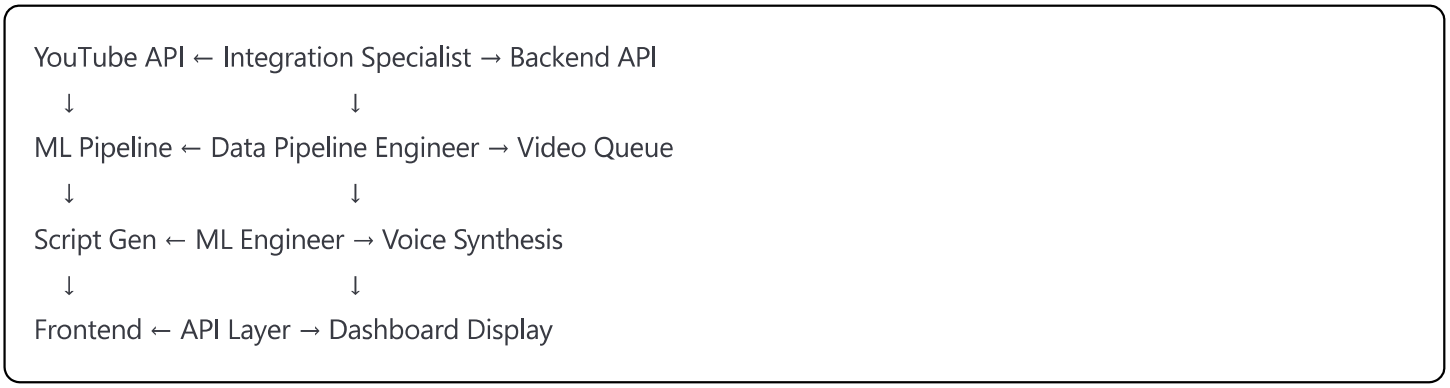
Appendix: Quick Reference

Key Metrics Dashboard

- **Videos Generated**: Target 1+, Stretch 10+

- **API Success Rate**: >95%
- **Cost per Video**: <$3.00
- **Pipeline Latency**: <10 minutes
- **Error Rate**: <5%

Critical Dependencies Map



Emergency Contacts

- **Infrastructure Issues**: Platform Ops Lead (on-call)
- **API Failures**: Integration Specialist
- **ML Pipeline**: AI/ML Team Lead
- **Production Issues**: DevOps Engineer (primary)

Document Version: 1.0
Sprint: Week 1 (Days 6-10)
Last Updated: Week 1, Day 1
Next Review: Friday 2:00 PM Sprint Demo
Owner: CTO/Technical Director