

# Work Order 18: Flow Overlay Report - Implementation Complete

---

## Executive Summary

**Successfully replaced the second motion video analyzer with a new Flow Overlay Report** that provides comprehensive momentum transfer analysis aligned with the BARRELS Flow System (Ground → Engine → Barrel).

## Key Achievements

- ✓ Replaced “Motion” tab with “Flow” tab in video detail page
  - ✓ Created Flow Report data structures for overlay, metrics, notes, and drills
  - ✓ Extended database schema to store flow analysis data
  - ✓ Built Flow Report API endpoint for generating coaching insights
  - ✓ Designed Flow Overlay Report UI with BARRELS styling (dark + gold accents)
  - ✓ Integrated with existing FLOW score and momentum transfer concepts
- 

## Implementation Overview

### Problem Solved

The original “Motion” tab showed a generic joint skeleton overlay that was disconnected from the BARRELS Flow System terminology. Users saw “Motion Analysis” but it didn’t align with:

- POWER / **FLOW** / CONTACT scores
- Ground → Engine → Barrel momentum transfer
- A-B-C timing markers (Trigger, Fire, Contact)

### Solution

Created a dedicated **Flow Overlay Report** that:

1. Shows skeleton overlay with A-B-C timing markers
  2. Displays key flow metrics (momentum, tempo ratio, smoothness, efficiency)
  3. Provides 2-3 coaching notes specific to momentum flow
  4. Recommends 1-3 drills to improve flow weaknesses
- 

## Technical Implementation

### 1. Data Model Updates

**File:** `prisma/schema.prisma`

**New Fields Added to Video Model:**

```
// Flow Overlay Report (W018)
flowOverlayMp4Url      String?    // S3 URL to flow overlay video
flowOverlayGifUrl       String?    // S3 URL to GIF preview
flowMomentumScore       Float?     // 0-100, overall momentum flow
flowTempoRatio          String?    // e.g. "2.8 : 1.0" (Load:Fire ratio)
flowSmoothnessScore     Float?     // 0-100, kinematic sequence
flowTransferEfficiency Float?     // 0-100, energy transfer
flowReportGenerated     Boolean   @default(false)
flowReportGeneratedAt   DateTime?
```

**Migration:** Database schema updated via `prisma db push` ✓

## 2. TypeScript Types

**File:** `lib/flow-report-types.ts`

**Core Interfaces:**

```
export interface FlowOverlayData {
  flowOverlayMp4Url: string | null;
  flowOverlayGifUrl: string | null;
  triggerFrame: number | null; // Frame A
  fireFrame: number | null; // Frame B
  contactFrame: number | null; // Frame C
}

export interface FlowMetrics {
  momentumFlowScore: number | null; // 0-100
  tempoRatio: string | null; // e.g. "2.8 : 1.0"
  flowSmoothnessScore: number | null; // 0-100
  transferEfficiencyScore: number | null; // 0-100
}

export interface FlowCoachingNote {
  id: string;
  text: string;
}

export interface FlowDrill {
  id: string;
  title: string;
  description: string;
}

export interface FlowReport {
  overlay: FlowOverlayData;
  metrics: FlowMetrics;
  notes: FlowCoachingNote[];
  drills: FlowDrill[];
}
```

**Helper Functions:**

- `hasFlowReportData(report)` : Checks if flow data is available
- `getFlowScoreColor(score)` : Returns traffic light color (green/yellow/red/gray)

### 3. Flow Report Generator

**File:** lib/flow-report-generator.ts

**Purpose:** Generates coaching notes and drill recommendations based on video analysis.

**Key Functions:**

1. `generateFlowReport(data: VideoAnalysisData): FlowReport`
  - Aggregates overlay data (URLs, frame markers)
  - Calculates or uses existing flow metrics
  - Generates 2-3 coaching notes
  - Suggests 1-3 drills based on weaknesses
  
2. `calculateTempoRatio(tagA, tagB, tagC): string`
  - Computes Load:Fire ratio from A-B-C tags
  - Example: "2.8 : 1.0" (load is 2.8x longer than fire)
  
3. `generateCoachingNotes(metrics): FlowCoachingNote[]`
  - **Overall Flow Assessment:** Based on `momentumFlowScore`
    - ≥85: "Excellent momentum transfer"
    - ≥70: "Solid fundamentals, focus on hip rotation"
    - <70: "Engine Flow needs attention, work on hip-torso separation"
  - **Tempo Ratio Feedback:** Based on `tempoRatio`
    - 3.0: "Load phase too long, be more explosive"
    - <2.0: "Tempo rushed, take a longer load"
    - 2.0-3.0: "Solid tempo range"
  - **Smoothness/Efficiency:** If significantly lower than overall score
  
4. `suggestFlowDrills(metrics): FlowDrill[]`
  - Low flow score → "Hip Rotation Isolations"
  - Low smoothness → "Step-Behind Soft Toss" (sequencing drill)
  - General → "Torque & Separation Drill"
  - Fallback → "Medicine Ball Rotational Throws"

### 4. API Endpoint

**File:** app/api/videos/[id]/flow-report/route.ts

**Endpoint:** GET /api/videos/[id]/flow-report

**Authentication:** Required (user session)

**Authorization:** User owns video OR admin/coach role

**Response:**

```
{
  "success": true,
  "flowReport": {
    "overlay": {
      "flowOverlayMp4Url": "https://s3.../overlay.mp4",
      "flowOverlayGifUrl": "https://i.ytimg.com/vi/fBYY08Zp9LQ/hqdefault.jpg",
      "triggerFrame": 45,
      "fireFrame": 90,
      "contactFrame": 120
    },
    "metrics": {
      "momentumFlowScore": 78,
      "tempoRatio": "2.6 : 1.0",
      "flowSmoothnessScore": 75,
      "transferEfficiencyScore": 80
    },
    "notes": [
      {
        "id": "flow-good",
        "text": "Your Engine Flow (hips → torso) shows solid fundamentals..."
      },
      {
        "id": "tempo-good",
        "text": "Your timing rhythm (load to fire) is in a solid range..."
      }
    ],
    "drills": [
      {
        "id": "hip-rotation-drill",
        "title": "Hip Rotation Isolations",
        "description": "Practice rotating your hips aggressively..."
      }
    ]
  }
}
```

#### Error Responses:

- 401 Unauthorized : No user session
- 403 Forbidden : User doesn't own video and is not admin/coach
- 404 Not Found : Video doesn't exist
- 400 Bad Request : Video not analyzed yet

**Database Update:** On first generation, saves metrics to `Video` record and sets `flowReportGenerated = true`.

---

## 5. Flow Overlay Report Component

**File:** `components/flow-overlay-report.tsx`

**Purpose:** Displays flow analysis in a visually engaging, BARRELS-styled UI.

**Structure:**

```
<FlowOverlayReport videoId={string} videoUrl={string} />
```

## Key Sections:

### 1. Header

- Gold Zap icon
- “Flow Overlay Report” title
- “Momentum Transfer Analysis” subtitle

### 2. Flow Overlay Video (if available)

- Video player with overlay clip
- A-B-C marker legend at bottom:
  - Green dot: A (Trigger)
  - Yellow dot: B (Fire)
  - Red dot: C (Contact)
  - Displays frame numbers for each marker

### 3. Flow Metrics Summary

- **Momentum Flow:** Color-coded badge (green/yellow/red)
  - Green  $\geq 85$ , Yellow  $\geq 70$ , Red  $< 70$
  - Label: “Ground → Engine → Barrel”
- **Tempo Ratio:** Gold text, e.g. “2.8 : 1.0”
- Label: “Load : Fire Duration”
- **Flow Smoothness:** Gray badge (0-100)
- Label: “Kinematic Sequence”
- **Transfer Efficiency:** Gray badge (0-100)
- Label: “Energy Transfer”

### 4. Flow Coaching Notes

- Numbered list (1, 2, 3) in gold circles
- Each note in a gray card with border
- Animated entry (fade + slide from left)

### 5. Recommended Flow Drills

- Numbered cards (gradient background)
- Bold title + detailed description
- Animated entry (fade + slide up)

## States:

- **Loading:** Spinner with gold color
- **Error:** Red alert with error message
- **No Data:** Gray activity icon + “No flow data available”
- **Success:** Full report display

## Styling:

- Dark background ( bg-gray-800/50 )
  - Gold accents for headers and icons
  - Traffic light colors for score badges
  - Responsive grid layout (1 col mobile, 2 col desktop)
-

## 6. Video Detail Page Integration

**File:** app/video/[id]/video-detail-client.tsx

**Changes:**

1. **Import:** Replaced `MotionTab` with `FlowOverlayReport`

```
typescript
import { FlowOverlayReport } from '@/components/flow-overlay-report';
```

2. **Tab State:** Changed `'motion'` to `'flow'`

```
```typescript
const [activeTab, setActiveTab] = useState<
  'overview' | 'flow' | 'breakdown' | 'drills' | 'history'
  ('overview');
```

```

3. **Tabs Array:** Renamed tab label

```
typescript
{ id: 'flow', label: 'Flow' }
```

4. **Tab Content:** Replaced `MotionTab` with `FlowOverlayReport`

```
tsx
{activeTab === 'flow' ? (
  <div className="space-y-6">
    <RickTip
      variant="compact"
      text="Watch your momentum transfer from Ground → Engine → Barrel.
        The overlay shows key timing points (A-B-C) and flow efficiency."
    />
    <FlowOverlayReport
      videoId={video.id}
      videoUrl={videoUrl || ''}
    />
  </div>
) : ...}
```

## Visual Design

### Color Palette

- **Background:** Dark gray (`bg-gray-800/50`, `bg-gray-900/30`)
- **Borders:** Dark gray (`border-gray-700`)
- **Primary Accent:** Gold (`text-barrels-gold`, `bg-barrels-gold/20`)
- **Text:** White for headings, gray for body (`text-white`, `text-gray-300`)
- **Traffic Light Colors:**
  - Green  $\geq 85$ : `text-emerald-400`, `bg-emerald-500/10`, `border-emerald-500/30`
  - Yellow  $\geq 70$ : `text-yellow-400`, `bg-yellow-500/10`, `border-yellow-500/30`
  - Red  $< 70$ : `text-red-400`, `bg-red-500/10`, `border-red-500/30`

## Icons

- **Zap**: Flow report header (momentum/electricity)
- **TrendingUp**: Flow metrics section
- **CheckCircle2**: Coaching notes section
- **Target**: Recommended drills section
- **Loader2**: Loading state
- **AlertCircle**: Error state

## Animations

- **Report Entry**: Fade + slide up ( opacity: 0 → 1 , y: 20 → 0 )
  - **Coaching Notes**: Staggered fade + slide from left (delay: index \* 0.1s )
  - **Drills**: Staggered fade + slide up (delay: index \* 0.15s )
- 

## Integration with Existing Systems

### 1. FLOW Score Alignment

The Flow Overlay Report is directly tied to the existing **FLOW** score in the BARRELS system:

- **Dashboard**: POWER / **FLOW** / CONTACT circular gauges
- **Video Detail**: BARREL Score breakdown with Engine Flow
- **Momentum Transfer Card**: powerFlow sub-score (now called “Engine Flow”)

The momentumFlowScore in the Flow Report uses:

1. flowMomentumScore (if explicitly set)
2. engine score (legacy field, represents Engine Flow)
3. overallScore (fallback)

### 2. A-B-C Timing Tags

The Flow Report leverages existing timing tags:

- **tagA**: Trigger/Load Start (seconds)
- **tagB**: Fire/Launch (seconds)
- **tagC**: Contact (seconds)

These tags are used to:

- Calculate tempoRatio (Load:Fire duration)
- Determine frame markers for overlay video
- Generate tempo-specific coaching notes

### 3. Terminology Consistency

#### Replaced:

- “Motion Analysis” → “Flow Overlay Report”
- “Joint flow” → “Momentum transfer”
- Generic skeleton overlay → Flow overlay with A-B-C markers

#### Maintained:

- Ground → Engine → Barrel (momentum flow path)
- POWER / FLOW / CONTACT (core scores)
- Engine Flow (formerly “powerFlow”)

# Testing Guide

## Manual Test Scenarios

### Test 1: Flow Report with Full Data

#### Setup:

1. Log in as a player with an analyzed video
2. Ensure video has:
  - `analyzed = true`
  - `engine score` (or `overallScore`)
  - Optional: `tagA`, `tagB`, `tagC` for timing

#### Steps:

1. Navigate to `/video/[id]`
2. Click "Flow" tab
3. Observe loading spinner
4. Verify flow report displays:
  - Flow metrics with scores
  - 2-3 coaching notes
  - 1-3 recommended drills
5. Check color coding:
  - Green badge for high scores ( $\geq 85$ )
  - Yellow for medium (70-84)
  - Red for low (<70)

#### Expected Results:

- Report loads within 2 seconds
- Metrics show realistic values (0-100)
- Notes are specific and actionable
- Drills have titles + descriptions
- Animations play smoothly

### Test 2: Flow Report without Overlay URLs

#### Setup:

1. Video has analysis data but no `flowOverlayMp4Url`

#### Steps:

1. Navigate to Flow tab
2. Verify overlay video section is **not displayed**
3. Verify metrics, notes, and drills **are displayed**

#### Expected Results:

- No video player shown
- Metrics grid still visible
- Coaching content still useful

### Test 3: Unanalyzed Video

#### Setup:

1. Video has `analyzed = false`

**Steps:**

1. Navigate to Flow tab
2. Observe error message

**Expected Results:**

- Error card displayed
- Message: "Video not analyzed"
- No crash or blank screen

**Test 4: Tempo Ratio Calculation****Setup:**

1. Video has `tagA = 0.5`, `tagB = 0.8`, `tagC = 1.0`
2. Load duration =  $0.8 - 0.5 = 0.3\text{s}$
3. Fire duration =  $1.0 - 0.8 = 0.2\text{s}$
4. Ratio =  $0.3 / 0.2 = 1.5$

**Steps:**

1. Navigate to Flow tab
2. Check "Tempo Ratio" metric

**Expected Results:**

- Shows "1.5 : 1.0"
- Label: "Load : Fire Duration"
- Coaching note mentions tempo (rushed in this case)

**Test 5: Authorization Check****Setup:**

1. User A owns a video
2. Log in as User B (not owner, not admin)

**Steps:**

1. Navigate to `/video/[videoId]` (User A's video)
2. Click Flow tab
3. Observe API response

**Expected Results:**

- 403 Forbidden error
- Error message displayed in UI
- No data leak

**Files Changed****New Files (4)**

1. `lib/flow-report-types.ts`
  - TypeScript interfaces for Flow Report
  - Helper functions for scoring and validation
2. `lib/flow-report-generator.ts`
  - Logic for generating coaching notes and drills

- Tempo ratio calculator
  - Smart drill recommendations
3. `app/api/videos/[id]/flow-report/route.ts`
- API endpoint for fetching flow report
  - Database integration
  - Authorization checks
4. `components/flow-overlay-report.tsx`
- React component for displaying flow report
  - BARRELS-styled UI with animations
  - Responsive design

## Modified Files (2)

1. `prisma/schema.prisma`
  - Added 8 new fields to `Video` model
  - Flow overlay URLs, metrics, generation status
2. `app/video/[id]/video-detail-client.tsx`
  - Replaced `MotionTab` import with `FlowOverlayReport`
  - Changed tab from “Motion” to “Flow”
  - Updated tab content rendering

## Removed Import (1)

1. `components/motion-tab.tsx`
    - No longer imported in video detail page
    - File still exists for reference but is unused
- 

## Deployment Checklist

### Pre-Deployment

- [x] TypeScript compilation passes (`yarn tsc --noEmit`)
- [x] Next.js build succeeds (`yarn build`)
- [x] Database schema updated (`prisma db push`)
- [x] No console errors in development
- [x] Flow tab visible in video detail page

### Post-Deployment

- [ ] Verify Flow tab accessible on production
  - [ ] Test API endpoint: `GET /api/videos/[id]/flow-report`
  - [ ] Confirm database writes for `flowReportGenerated`
  - [ ] Check mobile responsiveness
  - [ ] Monitor server logs for errors
-

## Known Limitations

---

### Phase 1 (Current Implementation)

#### 1. No Overlay Video Generation:

- Flow overlay URLs (`flowOverlayMp4Url`, `flowOverlayGifUrl`) are currently `null`
- Requires WO17 (or future work) to generate skeleton + A-B-C overlay videos
- **Workaround:** Flow report still works with metrics/notes/drills

#### 2. Frame Marker Calculation:

- Assumes 30 FPS for frame number conversion
- Real FPS should be used from `video.fps` field
- **Impact:** Frame numbers may be slightly inaccurate

#### 3. Static Drill Library:

- Drills are generated from a hardcoded list in `flow-report-generator.ts`
- Future: Link to actual `Drill` database records
- **Impact:** Limited drill variety

### Phase 2 (Future Enhancements)

#### 1. Real-Time Overlay Generation:

- Integrate with video processing pipeline
- Generate overlays on-demand or during analysis
- Store in S3 and link to `flowOverlayMp4Url`

#### 2. Personalized Drill Links:

- Query `prisma.drill` for real drill records
- Filter by `category: 'Engine Flow'`
- Include drill videos and step-by-step guides

#### 3. Coach Notes System:

- Allow coaches to add custom flow notes
- Store in `CoachNote` table
- Display alongside AI-generated notes

#### 4. Flow Comparison:

- Compare player's flow metrics to model swings
- Show delta in tempo ratio, smoothness, efficiency
- Suggest specific adjustments

---

## Success Metrics

### Technical

- Flow Report API loads in <2 seconds
- Zero TypeScript errors
- Zero console errors in browser
- Mobile responsive (320px - 1920px)

### User Experience

- Flow tab clearly labeled and discoverable

- Coaching notes are specific and actionable
- Drills are relevant to identified weaknesses
- Color coding (green/yellow/red) is intuitive

## Business

- [ ] Increased engagement on video detail page (measure after deploy)
  - [ ] Reduced “what do I work on next?” support tickets
  - [ ] Higher drill completion rate (future tracking)
- 

## Summary

### What Was Replaced

#### Before (Motion Tab):

- Generic joint skeleton overlay
- Disconnected from BARRELS Flow terminology
- No coaching or drill recommendations
- Not aligned with POWER/FLOW/CONTACT system

#### After (Flow Overlay Report):

- Momentum transfer analysis with A-B-C markers
- Aligned with Ground → Engine → Barrel flow path
- 2-3 coaching notes + 1-3 recommended drills
- Directly tied to FLOW score
- BARRELS-styled UI (dark + gold)

### Key Terminology Updates

Old Term	New Term
Motion Analysis	Flow Overlay Report
Joint flow	Momentum transfer
Motion tab	Flow tab
Generic skeleton	Flow overlay with A-B-C

### Integration Points

- POWER / **FLOW** / CONTACT scores
  - Engine Flow (formerly powerFlow)
  - A-B-C timing tags (tagA, tagB, tagC)
  - Momentum Transfer Card
  - Dashboard flow metrics
-

## Next Steps

---

### Immediate (Production)

1. Deploy to `catchbarrels.app`
2. Test Flow tab with real user videos
3. Monitor API response times and errors
4. Gather user feedback on coaching notes

### Short-Term (1-2 weeks)

1. **Generate flow overlay videos** (WO17 integration)
  - Add skeleton rendering with A-B-C markers
  - Upload to S3 and populate `flowOverlayMp4Url`
2. **Link to real drills**
  - Query `prisma.drill` instead of hardcoded list
  - Add drill thumbnails and videos

### Long-Term (1-2 months)

1. **Coach notes system**
    - Allow coaches to add custom flow feedback
    - Display alongside AI notes
  2. **Flow comparison**
    - Compare player vs. model swing flow
    - Show tempo ratio deltas
  3. **Progress tracking**
    - Track flow improvements over time
    - Show flow score trend graphs
- 

## Maintenance

---

### Updating Coaching Notes Logic

**File:** `lib/flow-report-generator.ts`

1. Edit `generateCoachingNotes()` function
2. Adjust score thresholds or messaging
3. Test with various score ranges

### Adding New Drills

**File:** `lib/flow-report-generator.ts`

1. Edit `suggestFlowDrills()` function
2. Add new drill objects with `id`, `title`, `description`
3. Update conditional logic for when to suggest

### Adjusting Tempo Ratio Thresholds

**File:** `lib/flow-report-generator.ts`

1. Find `ratio > 3.0` and `ratio < 2.0` checks

2. Update thresholds based on pro player data
  3. Update coaching messages to match new ranges
- 

## Support & Troubleshooting

---

### Common Issues

**Issue:** Flow tab shows “Video not analyzed”

- **Cause:** `video.analyzed = false`
- **Fix:** Analyze video first via Overview tab

**Issue:** Metrics show null values

- **Cause:** Video has no `engine` or `overallScore`
- **Fix:** Re-analyze video or check scoring engine

**Issue:** No overlay video displayed

- **Cause:** `flowOverlayMp4Url` is `null`
- **Fix:** Generate overlay video (WO17) or wait for future enhancement

**Issue:** Coaching notes are generic

- **Cause:** Limited score data or all scores in same range
  - **Fix:** Normal behavior; notes adapt to available data
- 

## Work Order 18 Status: COMPLETE

---

**Implementation:** 100%

**Testing:** Manual testing ready

**Documentation:** Complete

**Build Status:**  Passes

**Production Ready:**  Yes

**Deployed to:** Awaiting checkpoint save and deploy

---

### Questions or Issues?

Refer to this document or check:

- `lib/flow-report-generator.ts` for coaching logic
- `components/flow-overlay-report.tsx` for UI
- `app/api/videos/[id]/flow-report/route.ts` for API