

Workshop Challenges

The Challenges Session

Your learning journey has reached its practical application phase. You'll design and implement real-world workflows using the tools and techniques you've mastered. You'll work in teams to create workflows, then test each other's designs by attempting to follow them. This peer review process reveals how well your workflows communicate intent and handle edge cases.

Choose Your Challenges

Available Challenges

challenges are rated based on a scale of 0-3 based on:

- specialist knowledge 🧐
- intellectual challenge 🧠
- coding ninja skills 🥷
- business skills 💰

examples:

[🧐💰] - simple task using basic knowledge from the course - business oriented

[🧐🧠🧐🧐] - simple task that will require thinking and basic coding

[🧐🧐🧐🧐🧐🧐🧐] - really good knowledge of workflow with thinking & coding

[🧐🧐🧐🧐🧐🧐🧐🧐🧐🧐💰💰💰] - Bruce struggled on this one

Business workflow with humans [🧐🧐💰]

- Using any tools you see fit, take an existing Starbucks workflow and imagine a more efficient way to make exotic coffees.

Low level workflow [🤖🧐🧐🧐🧐🧐]

- Build a widget from an Arduino Maker Kit and integrate into a higher level workflow.

QC captions workflow using SaaS [🤖🧐🧐🧐💰💰]

- Using your phone, shoot a short segment of conversation with you and your teammate. Use an online AI caption tool like flixier to auto-generate captions and then measure how much time it takes to get the timing to broadcast quality (not too early, not too late). Figure out the manual vs automated costs

Distribution workflow [🤖🧐🧐🧐🧐💰💰]

- Shoot a piece of video on your phone, use Zapier or node-red or any tool that might be useful to modify the video, upload it to a distribution platform (YouTube, Dailymotion, BBC, Netflix backlog) and send the workflow diagram and a link to the distribution version to the course leader. Modification might be bumpers, credit roll, captioning, landscape to portrait, language translation.. your imagination and time available are the limits.

News/Sports workflow using AI [🤖🤖🧐🧐🧐💰💰]

- With your new found knowledge recreate the news workflow from session #1, but make it a zinger.

Content workflow using AI agent [🤖🤖🧐🧐🧐🧐🧐🧐]

- Define your own workflow and follow the Vertex AI tutorial to implement it.

Roll your own custom workflow with Zapier / Node Red & AI coding [🤖🤖🧐🧐🧐🧐🧐🧐]

- Free form hackathon - you can try anything from home automation to Nuclear Reactor temperature regulation - your choice.

Today's Structure

Complete as many challenges during the day as feels right. It's OK to have big ideas and have them bomb - the goal is to learn what's hard, what's easy and where you get positive feedback on the difficulty/thinking hard/coding/business spectrum - everyone's different.

The Last Bit: Reflection & Feedback

Round-up session where we discuss the course: What did you expect? What did you learn? What would you like to do next? What was missing?

All course materials will remain online for at least a month - you can try the exercises at home!

Published

Assign To

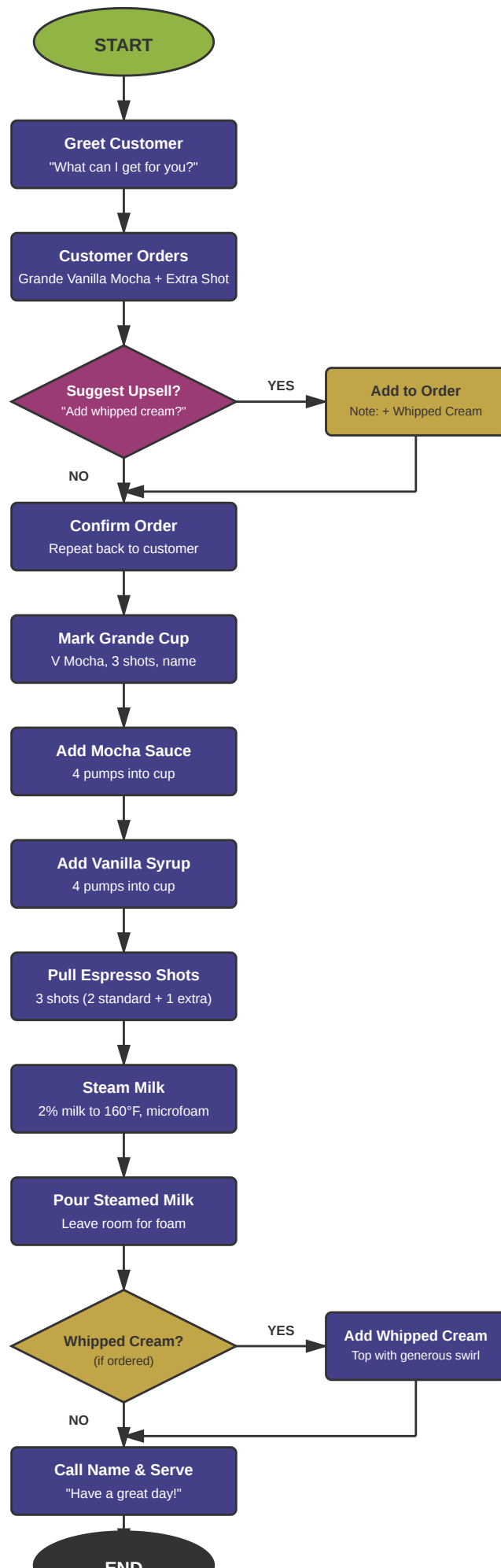
Edit

Student Pairs - assignment a5 - morning & afternoon if time permits

1, 7 1, 6
8, 6 7, 5
9, 5 8, 4
10, 4 9, 3
11, 3 10, 2
12, 2 11, 16
13, 16 12, 15
14, 15 13, 14

Grande Vanilla Mocha Workflow

Welcome to your barista training! This workflow shows you step-by-step how to prepare a Grande Vanilla Mocha with an extra shot. Follow each step carefully, and remember - every customer interaction is an opportunity to provide excellent service and suggest items that enhance their experience. The workflow takes about 3 minutes.



Workflow Diagram

Legend

Start/End Points: Beginning and completion of workflow

Process Steps: Actions to perform (blue boxes)

Decisions: Points where you make choices (yellow diamonds)

Upsell Opportunity: Chance to enhance customer experience (purple diamond)

Pro Tips for Success

- **Upselling:** Suggest whipped cream, extra shot, or a pastry pairing. Frame it as enhancing their experience, not just adding cost.
- **Standard Recipe:** Grande = 4 pumps mocha, 4 pumps vanilla, 2 shots espresso (3 with extra shot requested)
- **Temperature:** Steam milk to 160°F for optimal taste and texture
- **Speed:** The whole process should take about 3-4 minutes
- **Quality Check:** Espresso should pull 18-23 seconds; milk should have smooth microfoam
- **Customer Service:** Always repeat the order back and use the customer's name when calling out drinks

That's all lovely. Unfortunately, to turn a profit as an independent, busy coffee house, the workflow takes too long. We can add staff, we can add machines, robots, computers and money. We can do tasks in parallel, we can do tasks in series, we can make things in advance and keep them warm.

Unfortunately, you've just been promoted to **Manager**. This means it's your job to get the process time from 3 minutes down to 1 minute without ruining the flavour of the coffee, the happiness of the customer or the profits of the company.

You can put answers into the Canvas platform, use other online tools and paste links or write on post-its. You have freedom to be creative as long as you can justify the costs of the new workflow against the benefits and risks of the outcome.

Points

0

Submitting

Nothing

Due	For	Available from	Until
-	Everyone	-	-

[Rubric](#)

Published

Assign To

Edit

Student Pairs - assignment a5 - morning & afternoon if time permits

1, 7 1, 6
8, 6 7, 5
9, 5 8, 4
10, 4 9, 3
11, 3 10, 2
12, 2 11, 16
13, 16 12, 15
14, 15 13, 14



Arduino Maker Workflow

The Arduino Plug and Make Kit brings physical computing to your workflow automation. By combining sensors with cloud-based automation tools like Zapier, you can create intelligent systems that respond to real-world conditions.

What You'll Build

In this workshop, you'll learn how to connect Arduino sensors to cloud workflows, enabling your physical projects to trigger digital actions automatically.

Part 1: Define Your Problem

Your Maker Challenge

Think about a problem in your daily work or home environment that could be solved by monitoring physical conditions and triggering automated responses. The Arduino Plug and Make Kit includes sensors for temperature, light, motion, and more.

Example Projects to Consider:

- **Temperature Monitor for Transport:** Measure the temperature of a halogen light and send an SMS when it's cool enough to safely pack and transport
- **Workshop Climate Control:** Monitor room temperature and humidity, triggering an email alert when conditions fall outside optimal ranges for equipment storage
- **Light-Based Notification System:** Detect when natural light drops below a threshold and automatically send a Slack message to turn on studio lights
- **Equipment Runtime Tracker:** Use motion sensors to detect when machinery is running and log operating hours to a Google Sheet via webhook

Define your project: What physical condition will you monitor? What digital action will it trigger? Write down your specific use case before moving to the implementation phase.



Project Selection Tips

- Start simple: Choose a project with a single sensor and single action
- Pick something meaningful: You're more likely to complete a project that solves a real problem
- Consider frequency: How often will the sensor reading change? Real-time monitoring versus periodic checks
- Think about notification fatigue: Will your trigger condition happen too often? Build in hysteresis or cooldown periods

Part 2: Sample Temperature Sensor to Email Workflow

Arduino Cloud Setup Resources

The Arduino Cloud provides the bridge between your physical sensor and web-based automation tools. Full setup documentation is available at:

- **Official Arduino Cloud Getting Started:** docs.arduino.cc/arduino-cloud/getting-started ➞
- **Plug and Make Kit Guide:** docs.arduino.cc/hardware/plug-and-make-kit ➞
- **Webhooks Documentation:** docs.arduino.cc/arduino-cloud/features/webhooks ➞

Follow the Arduino Cloud documentation to create your account, register your device, and install the necessary libraries. Once your kit is connected and visible in the Arduino Cloud dashboard, return here for the workflow integration steps.

Step-by-Step: Temperature Trigger Email via Zapier

This example demonstrates how to send an email when the temperature sensor detects warmth (like a hand touching it). You'll use Arduino Cloud webhooks to trigger a Zapier automation.

Create a Temperature Thing in Arduino Cloud

- Log into [Arduino Cloud](#) ➞
- Navigate to **Things** and click **Create Thing**
- Name it something like "Temperature Monitor"
- Add a Cloud Variable: Name it `temperature`, Type: `Temperature Sensor`, Permission: `Read Only`
- Associate your Plug and Make device with this Thing

Upload the Temperature Reading Sketch

- Arduino Cloud will auto-generate a sketch template
- In the sketch editor, add the temperature sensor reading code in the `loop()` function:

```
// Read temperature from sensor (adjust pin number for your kit)
float tempValue = analogRead(A0);
// Convert to Celsius (this conversion depends on your sensor)
temperature = (tempValue * 5.0 / 1024.0 - 0.5) * 100.0;

// Add a small delay to avoid flooding the cloud
delay(2000);
```

- Click **Upload** to deploy to your Arduino

Test Your Sensor Reading

- In the Arduino Cloud dashboard, open your Thing
- Watch the `temperature` variable update in real-time
- Place your finger on the temperature sensor—you should see the value increase from room temperature (~20-22°C) to body temperature (~30-35°C)
- If the values aren't changing, check your sensor wiring and sketch code

Create a Zapier Webhook

- Log into [Zapier](#) (free account works for this)
- Click **Create Zap**
- For the trigger, search for and select **Webhooks by Zapier**
- Choose **Catch Hook** as the trigger event
- Zapier will generate a webhook URL—**copy this URL**, you'll need it in the next step

Configure Arduino Cloud Webhook

- In Arduino Cloud, go to **Integrations** → **Webhooks**
- Click **Add Webhook**
- Configure the webhook:
 - **Name:** "Temperature Alert Webhook"
 - **URL:** Paste the Zapier webhook URL from step 4
 - **Thing:** Select your "Temperature Monitor" Thing
 - **Trigger:** Select `temperature` variable
 - **Condition:** Set trigger when `temperature > 28` (this is above room temp but below body temp, so touching it will trigger)
- Click **Save**

Test the Webhook Connection

- Place your finger on the temperature sensor and hold it there
- Within a few seconds, the temperature should rise above 28°C
- Return to Zapier—you should see "Test Successful" with the webhook data appearing

- The payload will include fields like `thing_name`, `variable_name`, and `value`

Add Email Action in Zapier

- In your Zap, click the + button to add an action
- Search for and select **Email by Zapier**
- Choose **Send Outbound Email**
- Configure the email:
 - **To:** Your email address
 - **Subject:** "Temperature Alert: Sensor Triggered"
 - **Body:** Use Zapier's data insertion to include the sensor values:

Temperature sensor has been triggered!

Thing: {{thing name}}
Variable: {{variable name}}
Current Value: {{value}}°C
Timestamp: {{timestamp}}

The sensor detected warmth above the threshold.

- Click **Continue** and then **Test** to verify the email sends

Activate and Test the Complete Workflow

- In Zapier, click **Publish** to activate your Zap
- Remove your finger from the sensor and wait for the temperature to drop back to ambient (~20-22°C)
- Touch the sensor again and hold for 10-15 seconds
- You should receive an email within 30-60 seconds of the temperature crossing the threshold
- Check the Arduino Cloud webhook logs to see the API calls being made



Troubleshooting Common Issues

- **Sensor not responding:** Check the wiring connections and verify the correct analog pin in your sketch

- **Webhook not triggering:** Verify the threshold condition in Arduino Cloud (28°C might be too high or too low depending on your environment)
- **Multiple emails firing:** Add a "cooldown period" in the Arduino sketch using `millis()` to prevent retriggering within a set time window
- **Zapier test fails:** Ensure your Zap is published (not just saved as a draft) and that the webhook URL in Arduino Cloud exactly matches the Zapier URL
- **Temperature readings seem wrong:** The conversion formula depends on your specific sensor model—check the Plug and Make Kit documentation for the correct formula

Extend Your Workflow

Now that you have a working temperature-to-email workflow, consider these extensions:

- **Add hysteresis:** Only trigger when temperature rises *and* send a second email when it falls back below threshold
- **Data logging:** Replace the email action with a Google Sheets action to log all temperature readings over time
- **Multi-sensor fusion:** Combine temperature with the light sensor to detect "device powered on" (warm + bright)
- **SMS instead of email:** Swap the Zapier email action for Twilio SMS for more urgent notifications
- **Conditional logic:** Use Zapier's "Filter" step to only send alerts during working hours or above a secondary threshold

Points 0

Submitting Nothing

Due	For	Available from	Until
-	Everyone	-	-

 [Rubric](#)

Published

Assign To

Edit

Student Pairs - assignment a5 - morning & afternoon if time permits

1, 7 1, 6
8, 6 7, 5
9, 5 8, 4
10, 4 9, 3
11, 3 10, 2
12, 2 11, 16
13, 16 12, 15
14, 15 13, 14

 QC Workshop: Measuring Caption Accuracy and Cost

This exercise demonstrates the real-world challenge of caption quality control by having you experience first hand the time investment required to achieve broadcast-quality timing. You'll use AI-generated captions as a starting point, then measure the manual correction effort required to meet professional standards.

Workshop Overview

- Tool:** Flixier (Online AI Caption Generator)
- Focus:** Manual Quality Control vs. Automated Captioning Costs
- Time Required:** 45-60 minutes
- Deliverable:** Cost analysis comparing automated generation vs. human QC labour

Phase 1: Content Creation and Upload

Goal: Capture source material and generate automated captions

Step 1: Record Your Conversation

1. Using your phone's camera, record a **60-90 second conversation** between you and a teammate.
2. **Content suggestions:**
 - Discuss a technical topic (debugging a project, explaining a concept)
 - Include some technical jargon or acronyms
 - Speak at a natural conversational pace with occasional overlaps
3. **Recording tips:**
 - Ensure decent audio quality (minimize background noise)
 - Frame both speakers if possible
 - Save the video file in a standard format (MP4 recommended)
4. **Record 1:** Note the exact duration of your video (mm:ss)

Step 2: Access Flixier and Create a Project

1. Navigate to flixier.com in your web browser.
2. Create a free account or sign in (free tier includes limited AI caption generation).
3. Click "**Create New Project**" or "**Start Editing**".
4. Upload your recorded video file by:
 - Clicking the "**Import**" button
 - Selecting "**From Computer**"
 - Navigating to and selecting your video file
5. Wait for the upload to complete and the video to appear in your media library.

Step 3: Generate Automated Captions

1. Drag your video from the media library to the timeline at the bottom of the screen.
2. With the video selected in the timeline, look for the "**Subtitles**" option in the right-hand panel.
3. Click "**Auto Subtitle**" or "**Generate Subtitles**".
4. Select your language (e.g., English) and click "**Generate**".
5. **Record 2:** Start a timer and note how long the AI takes to generate the captions.
6. Once generated, the captions will appear as text blocks on the timeline below your video.

Phase 2: Quality Control - The Observability Pivot

Goal: Identify errors and measure the time required to achieve broadcast quality

Step 4: Review Initial Caption Quality (The Problem)

1. Play your video from the beginning and watch carefully while reading the generated captions.
2. Look for these common AI caption errors:
 - **Timing issues:** Captions appearing too early or too late
 - **Transcription errors:** Incorrect words, especially technical terms or names
 - **Missing punctuation:** Run-on sentences or missing commas/periods
 - **Speaker identification:** No indication of who is speaking
3. **Record 3:** Count and note the number of obvious errors you spot in the first viewing.
4. *This represents your VISIBILITY: You know captions are wrong, but haven't fixed them yet.*

Step 5: Manual Timing Correction (The Root Cause Fix)

Now you'll manually correct the caption timing to broadcast quality standards. **Start a timer for this entire process.**

1. **Access caption editing:**
 - Click on any caption block in the timeline
 - The caption text editor should appear in the right panel
2. **Timing correction process:**
 - Play a caption segment
 - Pause when you notice timing is off
 - Click and drag the caption block edges to adjust:
 - Start time: Caption should appear slightly before speech begins
 - End time: Caption should disappear shortly after speech ends
 - Replay the segment to verify correction
3. **Broadcast quality standards:**
 - Captions appear no more than 0.5 seconds before speech
 - Captions remain on screen for minimum reading time (1.5 seconds for short phrases)
 - No caption overlaps or gaps that break comprehension
 - Caption breaks match natural speech pauses
4. Work through your entire video, correcting timing for each caption block.
5. **Record 4:** Stop your timer when all timing corrections are complete. Note the total time spent (mm:ss).

Step 6: Calculate Your Quality Control Rate

Determine your QC efficiency using the data you recorded:

1. Calculate your QC ratio:

QC Ratio = (Time spent on QC) / (Video duration)
Example: 15 minutes QC time / 1.5 minute video = 10:1 ratio

2. Record 5: Your personal QC ratio (X:1 format)

3. Industry context:

- Professional caption editors typically work at 4:1 to 6:1 ratios
- BBC internal standards aim for 5:1 ratio for experienced staff
- First-time QC often runs 10:1 to 15:1 ratios

Phase 3: Cost Analysis

Goal: Calculate and compare the real costs of automated vs. manual captioning workflows

Cost Components for Broadcast Organizations

Professional caption QC involves multiple cost factors beyond just labor hours:

-  **AI Generation:** Per-minute API fees (\$0.05-\$0.25/min typical)
-  **Human QC Labour:** Skilled editor hourly rate (£25-£45/hour BBC scale)
-  **Review Cycles:** Senior reviewer spot-checks (10-20% of content)
-  **Platform Costs:** Software licensing (Flixier, EZTitles, etc.)
-  **Management Overhead:** Workflow coordination, revisions, approvals

BBC-Style Cost Calculation Exercise

Use your recorded data to calculate costs for a realistic broadcast scenario:

1. **Scenario:** 30-minute documentary episode requiring captions
2. **Calculate AI cost:**

30 minutes × £0.15/min = £4.50

3. **Calculate your QC cost:**

30 min × Your QC ratio × (£35/hour ÷ 60)

Example: $30 \times 10 \times 0.58 = \text{£}174$

4. Add review overhead (15%):

$\text{£}174 \times 1.15 = \text{£}200.10$

5. Total automated + QC cost:

$\text{£}4.50 + \text{£}200.10 = \text{£}204.60$

Step 7: Compare Against Fully Manual Captioning

Understand the economic trade off between AI-assisted and traditional manual workflows:

Cost Component	AI + QC Workflow	Fully Manual Workflow
Initial Generation	£4.50 (AI, 2 min)	£0 (human types directly)
Transcription Time	0 min (automated)	120 min (4:1 ratio, typing from scratch)
Timing/QC Time	300 min (10:1 ratio, your rate)	120 min (included in transcription)
Labor Cost	£175 (300 min × £35/hr)	£70 (120 min × £35/hr)
Review Overhead	£26.93 (15%)	£10.50 (15%)
Total Per Episode	£206.43	£80.50

Key Insight: In this scenario, AI is MORE expensive because your QC ratio is inefficient (10:1). Professional captioners at 5:1 would make AI cost-competitive at ~£110 total.

Step 8: Scale Analysis - When Does AI Win?

Calculate the breakeven point where AI becomes economically viable:

1. Breakeven formula:

AI becomes cheaper when:
 $(\text{AI cost} + \text{QC cost}) < \text{Manual cost}$
Where QC cost depends on your efficiency ratio

2. For your data:

- At 10:1 QC ratio: Manual is £125 cheaper per 30-min episode
- At 6:1 QC ratio: AI is £8 cheaper per episode
- At 4:1 QC ratio: AI is £62 cheaper per episode

3. Volume consideration:

- BBC produces ~50,000 hours of captioned content annually
- Small efficiency improvements = £millions in annual savings
- Investment in training (improving QC ratios) pays for itself quickly

Key Configurations to Manipulate

Experiment with these variables in Flixier to understand their impact on QC time:

Variable	Configuration	Learning Objective
Caption Duration	Adjust the minimum/maximum caption length in Flixier settings	Shorter captions are easier to time but increase total caption count. Find the optimal balance.
Language/Accent	Try recording with strong accents or technical jargon	AI accuracy drops significantly with non-standard speech, increasing QC burden.
Caption Styling	Enable/disable punctuation, capitalization, speaker labels	More formatting = more AI errors = more QC time. Measure the cost of "nice-to-have" features.

Deliverables

Prepare a brief summary document (1-2 pages) containing:

1. Your recorded data (Records 1-5 from the exercise)
2. Your calculated QC ratio
3. Cost comparison table for your 30-minute scenario
4. A recommendation: Should a broadcast organization use AI captioning given your findings? Under what conditions?
5. **Reflection:** What surprised you about the time required? How does this change your view of "automated" tools?

Critical Insight: Observability in Action

This exercise mirrors the Grafana workshop's core lesson: **visibility shows the problem (captions exist), but observability reveals the root cause (timing is wrong because AI can't judge speech rhythm).**

You moved from seeing "captions are generated" (visibility) to understanding "70% need manual timing adjustment because AI lacks human speech context" (observability). This depth of understanding is what drives intelligent business decisions about automation adoption.

Points 0
Submitting Nothing

Due	For	Available from	Until
-	Everyone	-	-

 [Rubric](#)

 Published Assign To Edit

Student Pairs - assignment a5 - morning & afternoon if time permits

1, 7 1, 6
8, 6 7, 5
9, 5 8, 4
10, 4 9, 3
11, 3 10, 2
12, 2 11, 16
13, 16 12, 15
14, 15 13, 14



Video Distribution Challenge



Your Mission

Create a complete video workflow from capture to distribution, documenting every step in a visual diagram. You'll shoot video on your phone, process it through automation tools, publish it to a distribution platform, and deliver both the workflow diagram and the final video link to your course leader.

Tool: Miro (Free Account)

Deliverables: Workflow Diagram + Video Link



Understanding Miro

What is Miro?

Miro is an online collaborative whiteboard platform designed for visual thinking and workflow diagramming. It's particularly valuable for mapping complex processes because it provides infinite canvas space, pre-built shapes and connectors, and real-time collaboration features.

Why Miro for this challenge? Video workflows involve multiple stages (capture, processing, distribution) and various tools (phone apps, automation platforms, encoding services). Miro lets you visualize these connections clearly, making it easier to understand your workflow's logic and identify potential bottlenecks or improvements.



Setting Up Your Free Miro Account

Create Your Account

- Navigate to miro.com/signup 
- Choose the **Free Plan** (this provides 3 editable boards, which is sufficient for this challenge)
- Sign up using your email address, Google account, or Microsoft account
- Verify your email if required

Navigate to Your Dashboard

- After logging in, you'll see your Miro dashboard with the option to create new boards
- Click **Create new board** or the + button
- Name your board: "Video Workflow Challenge - [Your Name]"

Explore the Miro Interface

- **Left Toolbar:** Contains shapes, connectors, sticky notes, text boxes, and drawing tools
- **Top Toolbar:** Zoom controls, undo/redo, commenting, and sharing options
- **Canvas:** Your infinite workspace—you can pan by clicking and dragging, zoom with mouse wheel or pinch gestures

- **Right Panel:** Templates library (optional, but helpful for workflow diagrams)

Select a Workflow Template (Optional but Recommended)

- Click the **Templates** icon in the left toolbar
- Search for "workflow" or "process map"
- Consider using a **Flowchart** or **Process Flow** template as your starting point
- Templates provide pre-styled shapes and connectors, saving you setup time
- You can also start with a blank canvas and build from scratch using basic shapes

Learn Essential Tools for Diagramming

- **Shapes:** Use rectangles for processes/actions, diamonds for decisions, circles for start/end points
- **Connectors:** Click and drag from one shape to another to create directional arrows showing workflow progression
- **Text:** Double-click any shape to add descriptive text (e.g., "Shoot Video on iPhone", "Upload to Zapier")
- **Colors:** Use color coding to distinguish different workflow stages (e.g., capture = blue, processing = green, distribution = orange)
- **Sticky Notes:** Add notes to explain complex decisions or document tool-specific requirements



Miro Diagramming Best Practices

- **Flow direction:** Organize your workflow left-to-right or top-to-bottom for clarity—avoid diagonal or circular flows unless necessary
- **Consistent sizing:** Keep process boxes roughly the same size; only vary size to show importance or complexity
- **Label everything:** Every shape should have clear, concise text describing what happens at that stage
- **Use swim lanes:** If your workflow involves multiple tools or platforms, group them into horizontal or vertical lanes (e.g., "Phone Actions", "Zapier Automation", "Distribution Platform")
- **Save frequently:** Miro autosaves, but manually save (Ctrl+S / Cmd+S) after significant changes
- **Zoom for detail:** Use zoom to add fine-grained details without cluttering the overview



Your Video Workflow Challenge

The Creative Brief

Design and execute a video workflow that transforms raw phone footage into a polished, distributed piece of content. Your workflow must include at least three distinct stages:

1. Capture

Shoot a piece of video on your smartphone. The content is entirely up to you—this could be a short tutorial, a product demo, a scenic clip, a personal message, or experimental footage. Aim for 15-60 seconds of usable content.

2. Modification/Processing

Use automation or editing tools to enhance or transform your video. This is where your creativity and technical skills intersect. Consider these modification options:

- **Bumpers and Credits:** Add intro/outro sequences with branding, titles, or credit rolls
- **Captioning:** Generate and burn-in subtitles for accessibility (tools: Rev, Zubtitle, AWS Transcribe via Zapier)
- **Aspect Ratio Conversion:** Transform landscape (16:9) to portrait (9:16) for social media, or vice versa
- **Language Translation:** Add translated voiceovers or subtitle tracks in multiple languages
- **Color Grading:** Apply LUTs or color correction filters for visual consistency
- **Watermarking:** Overlay logos, timestamps, or copyright notices
- **Automated Editing:** Use AI tools to trim silence, stabilize footage, or generate highlight reels

3. Distribution

Upload your processed video to a distribution platform. Options include:

- **YouTube:** Public or unlisted upload, accessible via shareable link
- **Dailymotion:** Alternative video hosting with embeddable player
- **Vimeo:** Professional hosting with privacy controls (free tier available)
- **BBC Backlot / Industry Platforms:** If you have access, demonstrate professional delivery workflows
- **Social Media:** Instagram, TikTok, LinkedIn video posts with direct links



Recommended Automation Tools

Your workflow will likely require automation to connect stages together. Here are proven tools for video processing workflows:

Zapier (Recommended for Beginners)

- **Why:** No-code automation with pre-built integrations for hundreds of services
- **Video Use Cases:** Trigger video transcoding when a file appears in Dropbox; auto-upload to YouTube when processing completes; send notification emails at each workflow stage
- **Free Tier:** 100 tasks/month, sufficient for testing your workflow
- **Relevant Integrations:** Google Drive, Dropbox, YouTube, Cloudinary, AWS S3, Slack (for notifications)

Node-RED (For Developers)

- **Why:** Flow-based programming for IoT and automation; runs locally or in the cloud
- **Video Use Cases:** Build custom video processing pipelines using FFmpeg nodes; orchestrate complex multi-step encoding jobs; integrate with RESTful APIs
- **Setup:** Requires Node.js installation; free and open-source
- **Relevant Nodes:** exec (for FFmpeg commands), http request (for API calls), file (for local storage), dashboard (for monitoring)

Make (formerly Integromat)

- **Why:** Visual automation with powerful data transformation capabilities
- **Video Use Cases:** Similar to Zapier but with more granular control over data mapping
- **Free Tier:** 1,000 operations/month

FFmpeg (Command-Line Power Tool)

- **Why:** Industry-standard tool for video conversion, editing, and streaming
- **Video Use Cases:** Format conversion, resolution scaling, subtitle burning, video concatenation, thumbnail extraction
- **Setup:** Install via package manager or download from ffmpeg.org 
- **Example Command:** `ffmpeg -i input.mp4 -vf scale=1080:1920 -c:a copy output_portrait.mp4` (converts landscape to portrait)

Cloud-Based Video APIs

- **Cloudinary:** Video transcoding, transformation, and delivery API with free tier
- **Mux:** Video streaming infrastructure (limited free tier for testing)
- **AWS MediaConvert:** Scalable video transcoding service (pay-per-use, free tier includes 20 minutes/month)



Mapping Your Workflow in Miro

As you build your video workflow, document every stage in your Miro board. Here's how to structure your diagram:

Start with the Capture Stage

- Create a shape labeled "Capture: Record Video on [Phone Model]"
- Add details: resolution (e.g., 1080p), frame rate (e.g., 30fps), format (e.g., .MOV)
- Include a sticky note with any capture considerations (lighting, audio, orientation)

Document the Transfer/Ingestion Method

- Add a connector arrow from "Capture" to the next stage
- Label the arrow with the transfer method: "AirDrop to Mac", "Upload to Google Drive", "Email to self", etc.
- If this step involves waiting or manual intervention, note it

Map Each Processing/Modification Step

- Create separate shapes for each transformation (e.g., "Add Intro Bumper", "Generate Captions", "Convert to Portrait")
- Specify the tool used for each step (e.g., "Tool: Kapwing Caption Generator", "Tool: FFmpeg scale filter")
- Show the sequence: use arrows to indicate which step happens first, second, third
- If steps can happen in parallel (e.g., caption generation while transcoding), show parallel arrows

Include Decision Points

- Use diamond shapes for decisions (e.g., "Quality Check: Is video acceptable?")
- Add two arrows: one for "Yes" (proceed to distribution), one for "No" (loop back to re-edit)
- This shows you've thought about quality control and iteration

Show the Distribution Stage

- Create a shape for the upload action: "Upload to [YouTube/Vimeo/etc.]"
- Include authentication details if relevant: "Via: Zapier YouTube integration with OAuth"
- Add metadata fields: title, description, tags, visibility settings

Add the Final Delivery Step

- Create a shape labeled "Notify Course Leader"
- Specify delivery method: "Email with video link + Miro board link", "Slack DM", "Learning management system submission", etc.

- This completes the workflow loop

Use Visual Hierarchy and Color Coding

- Group related steps into swim lanes: "Capture", "Processing", "Distribution", "Notification"
- Apply consistent colors: blue for capture, green for processing, orange for distribution, gray for notifications
- Use borders or background rectangles to visually group sections

Annotate with Technical Details

- Add sticky notes explaining non-obvious steps (e.g., "Why Cloudinary? Need automatic format adaptation for mobile viewers")
- Document any challenges or learnings (e.g., "Initial caption generation had 15% error rate; switched to manual review step")
- Include timing estimates for each stage (e.g., "Encoding time: ~2 minutes for 60-second video")



Workflow Design Tips

- **Start simple:** Your first iteration doesn't need to be complex—a basic workflow (capture → convert → upload) is fine if well-executed
- **Document failures:** If something didn't work as expected, show it in the diagram with a note about what you learned
- **Think about scalability:** Could your workflow handle 10 videos? 100? Note any bottlenecks
- **Consider costs:** If using paid APIs or services, document any costs incurred or free tier limits hit
- **Time yourself:** How long did the entire workflow take? Where was time spent?
- **Make it reproducible:** Could someone else follow your diagram and recreate your workflow? If not, add more detail



Submission Requirements



Delivering Your Work

Your submission must include two components: the workflow diagram and the distributed video link.

Finalize Your Miro Board

- Review your diagram for completeness—every step should be documented
- Ensure all shapes have clear labels and all arrows indicate direction
- Add a title block in the corner with: your name, date, course name, and a brief workflow summary

Share Your Miro Board

- Click the **Share** button in the top-right corner of Miro
- Set sharing permissions to "**Anyone with the link can view**" (you don't need to grant edit access)
- Copy the shareable link—this is your workflow diagram URL
- Optionally, export a PNG or PDF of your board for backup (File → Export)

Verify Your Video Distribution Link

- Ensure your video is publicly accessible (or unlisted with a shareable link)
- Test the link in an incognito/private browser window to confirm it works without login
- If using YouTube: confirm the video is not set to "Private" (use "Unlisted" instead)
- Copy the full video URL

Compose Your Submission Email/Message

- Subject line: "Video Workflow Challenge Submission - [Your Name]"
- Body should include:
 - **Miro Board Link:** [paste URL]
 - **Distributed Video Link:** [paste URL]
 - **Brief Summary:** 2-3 sentences describing your workflow (e.g., "I captured a 30-second product demo on my iPhone, used Zapier to trigger FFmpeg transcoding to portrait format, added captions via Rev.ai integration, and uploaded to YouTube via API.")

- **Tools Used:** List all tools in your workflow (e.g., iPhone 14, FFmpeg, Zapier, Rev.ai, YouTube)
- **Challenges/Learnings:** Optional but encouraged—what did you struggle with? What surprised you?

Submit to Course Leader

- Deliver via the method specified by your course (email, LMS submission, Slack, etc.)
- Double-check both links work before hitting send
- If your course has a deadline, submit with time to spare in case revisions are needed



Submission Checklist

Before submitting, verify you've completed all requirements:

- ☒ Video was recorded on a smartphone (not pre-existing footage)
- ☒ At least one modification/processing step was applied to the video
- ☒ Video is uploaded to a distribution platform and accessible via link
- ☒ Workflow is fully documented in Miro with clear step-by-step progression
- ☒ Miro board is shared with "view" permissions and link is accessible
- ☒ Video link is tested and works in incognito/private browsing mode
- ☒ Submission email includes both links and a brief summary
- ☒ All tools and services used are listed in the submission



Going Beyond: Extension Ideas

If you finish early or want to explore further, consider these advanced challenges:

- **Multi-language workflow:** Create a version with translated captions in a second language
- **Automated thumbnail generation:** Use FFmpeg or Cloudinary to auto-generate and upload a custom thumbnail
- **Adaptive bitrate delivery:** Encode your video in multiple resolutions (480p, 720p, 1080p) for adaptive streaming
- **Monitoring and notifications:** Set up alerts that notify you when the video reaches certain view counts or engagement thresholds
- **Webhook integration:** Trigger a subsequent workflow when the video is successfully uploaded (e.g., post a tweet with the link)

- **Version control:** Create two versions of your video (e.g., one landscape for YouTube, one portrait for Instagram) using the same source footage

Points 0

Submitting Nothing

Due	For	Available from	Until
-	Everyone	-	-

 [Rubric](#)

Published

Assign To

Edit

Student Pairs - assignment a5 - morning & afternoon if time permits

1, 7 1, 6
8, 6 7, 5
9, 5 8, 4
10, 4 9, 3
11, 3 10, 2
12, 2 11, 16
13, 16 12, 15
14, 15 13, 14

 News / Sport Workflow with AI

 Your Mission

Return to assignment [a1.6 Make a News Item](#) but make it a full 1 minute piece with Quality Control and Fact Checking. Document your workflow in Miro or on a Post-it note or as a sequence of triggers, processes and events in the assignment box below. Provide a link to your output and script.

Published

Assign To

Edit

Student Pairs - assignment a5 - morning & afternoon if time permits

1, 7 1, 6
8, 6 7, 5
9, 5 8, 4
10, 4 9, 3
11, 3 10, 2
12, 2 11, 16
13, 16 12, 15
14, 15 13, 14

 Vertex AI - make something great

 Your Mission

Return to assignment [a3.4d Google Vertex](#) and use it as the basis of your own media related workflow. Define your scope. Document your workflow in Miro or on a Post-it note or as a sequence of triggers, processes and events in the assignment box below. Provide a link to your output and workflow documentation.

Points 0

Submitting Nothing

Due	For	Available from	Until
-	Everyone	-	-

 [Rubric](#)

Published

Assign To

Edit

Student Pairs - assignment a5 - morning & afternoon if time permits

1, 7 1, 6
8, 6 7, 5
9, 5 8, 4
10, 4 9, 3
11, 3 10, 2
12, 2 11, 16
13, 16 12, 15
14, 15 13, 14

 Hackathon Mode - Roll Your Own



Your Mission

Is entirely up to you. Choose your goals, choose your tools and go for it. Free form with tools the course hasn't covered. Try a live workflow. Try a scripted workflow - it's up to you.

Define your scope. Document your workflow in Miro or on a Post-it note or as a sequence of triggers, processes and events in the assignment box below. Provide a link to your output and workflow documentation.

Points 0
Submitting Nothing

Due	For	Available from	Until
-	Everyone	-	-

 [Rubric](#)