

VERTICAL SCREENCAST / 9:16 / PHONE-FIRST

How to record technical demos for DevRel and software teams

A repeatable 9:16 field checklist — from staging the demo through the archived master.

34 checks · 5 stages · 1080×1920 · fillable + printable

CAPTURE 9:16 native, 1080×1920, punch-in room preserved.	CLARITY Oversized code, close mic, protected safe zones.	CONTROL Staged demo, marked takes, verified export.
--	--	---

THE STANDARD

Record for the phone from the beginning. Narrow capture, oversized code, close audio, protected safe zones, and a pre-staged demo beat a rescue edit every time.

PROJECT / SERIES

OWNER

RECORDING DATE

01 before the first take 02 during capture 03 before you publish

Use this sheet three times. If a box is still empty at the end, the video is not ready to ship.

01-03 / BUILD THE SOURCE

Make the footage easy to use

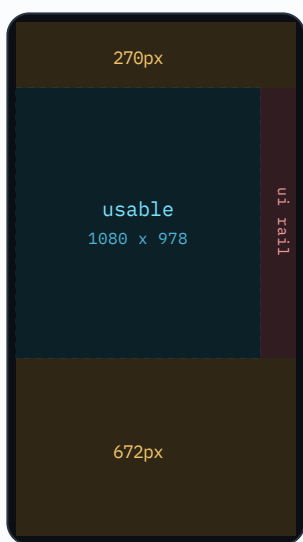
The edit should clarify the idea — not rescue the recording. These three passes protect the source before it becomes a post-production problem.

01 / before you record

6 CHECKS

Stage the environment so the take is the only variable.

- ☐ A one-sentence promise is written: after this, the viewer can _____.
- ☐ The first two seconds show the result, the error, or the before / after.
- ☐ One idea is selected. The clip is not trying to be a conference talk.
- ☐ Dependencies, servers, test data, API keys, and browser tabs are ready.
- ☐ Notifications, personal tabs, autocomplete, and secrets are hidden.
- ☐ A checkpoint or git branch exists for every demo state you may revisit.



1080 X 1920

SAFE-ZONE RULE

For a 1080×1920 master, keep essential content **below roughly 270px from the top, above roughly 672px from the bottom**, and clear of the right-side interaction rail.

Captions, logos, and the code you actually want read all live inside the cyan region. The usable center is smaller than it looks.

02 / capture setup

8 CHECKS

Set the frame, the type size, and the mic before the first take.

- ☐ Final format is chosen: native 9:16, or a deliberately narrow capture region.
- ☐ Target export is 1080×1920 MP4; source resolution leaves punch-in room.
- ☐ Editor font is large — start around 18–24px equivalent for vertical.
- ☐ Code wraps at roughly 40–55 characters; minimap and extra panels are hidden.
- ☐ Important code, captions, and logos stay inside the central safe region.
- ☐ Cursor is visible and moves deliberately; zooms are saved for the lines that matter.
- ☐ Mic is 4–6 inches away, slightly off-axis, with fans and HVAC under control.
- ☐ A ten-second audio test is recorded and played back before the real take.

03 / during recording

6 CHECKS

Record in sections. Protect the clean take.

- ☐ Show the payoff before explaining the setup.
- ☐ Record short matched sections instead of one long continuous take.
- ☐ Move the cursor with intent; pause before changing the screen state.
- ☐ If a sentence goes wrong, pause and redo it — do not bury the clean take.
- ☐ Read commands, package names, and API terms exactly as they appear.
- ☐ Mark a good take immediately so nobody has to rediscover it in the edit.

WHY THIS STAGE PAYS

Most vertical screencasts fail before the record button. A staged environment, one idea, and a written promise remove almost every reason to reshoot.

04-05 / FINISH AND SHIP

Protect the viewer's attention

A vertical technical video succeeds when the viewer can read the screen, understand the point, and keep listening. The final pass removes friction — it does not add decoration.

04 / edit + captions

7 CHECKS

Remove friction. Do not add decoration.

- ☐ Dead air, false starts, and duplicate takes are removed before polish begins.
- ☐ Zooms are purposeful — usually 2–3× for the code moment that needs attention.
- ☐ Captions are burned in above the bottom platform UI reserve, not in the lower third.
- ☐ Every command, product name, library, and technical term is checked manually.
- ☐ Caption contrast survives a phone screen and muted playback.
- ☐ Voice is clear and consistent; enhancement is rescue, not a substitute for capture.
- ☐ File names and versions make the handoff obvious to the next person.

COMMON FAILURE MODES

■ Captions in the lower third

Sits behind the platform UI on every major app. Raise them above the bottom reserve.

■ One long continuous take

A single flub costs the whole section. Record short, matched pieces instead.

■ Zooms used as decoration

Movement without a reason reads as noise. Zoom only where the code changes.

■ Default editor font size

Legible on a 27-inch display, unreadable on a phone held at arm's length.

■ Enhancement as rescue

Processing cleans noise. It cannot move a distant mic closer after the fact.

■ Shipping without a muted watch

Most of the feed plays silently. If it fails muted, it fails.

05 / final gate

7 CHECKS

Watch it the way the viewer will. Then archive it.

- ☐ Export is a clean 1080×1920 MP4 with no platform watermark.
- ☐ The first two seconds make sense without context and without sound.
- ☐ The export is watched on a phone once muted, once with sound.
- ☐ Code stays legible after compression; no critical content is hidden by UI.
- ☐ Captions, logo, picture-in-picture, and CTA clear the safe zones.
- ☐ The description and call to action point to the next useful step.
- ☐ Clean master, project file, captions, and notes are archived together.

ARCHIVE PACKAGE

clean master – 1080x1920 mp4
project file + source bin
caption source + burned-in export

take notes and marked selects
thumbnail / cover frame
this checklist, completed

SHIP GATE

If a box is still empty, the video is not ready to ship. Sign off below only when every stage is closed — this sheet travels with the project file.

APPROVED BY

EXPORT VERSION

SHIP DATE

If the work cannot scale, it is not finished.

WHAT HAPPENS AFTER THE CHECKLIST

One clip is a task. A launch is a system.

This checklist keeps a single vertical screencast honest. TCV Studio builds the system around it — so a launch ships every variant, every channel, and every cutdown from one source of truth, on your cadence.

AI-FIRST PIPELINE

Ingest, classify, version, and adapt — with human review gates where quality is decided.

SCALABLE VARIANTS

One master creative, customized by audience, channel, and aspect ratio.

TECHNICAL NARRATIVE

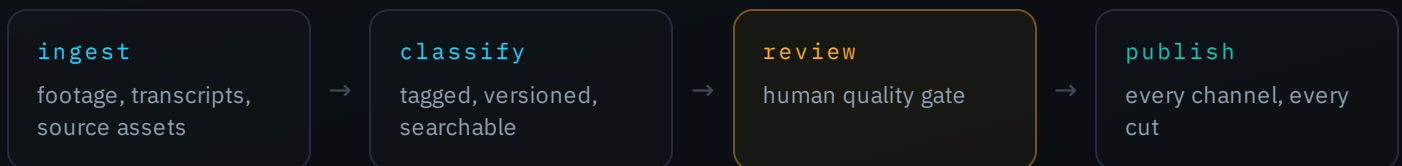
We translate technical products without flattening the nuance.

NEXT STEP

Audit your content pipeline with us.

hello@tcv.studio

THE PRODUCTION SYSTEM, PLAINLY



ai accelerates versioning and adaptation · human review protects quality