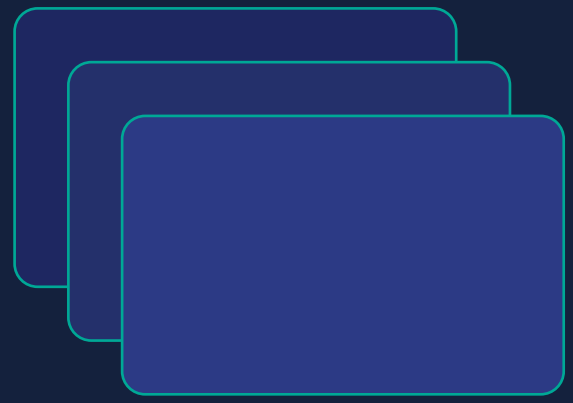


UI Sandboxes

A better front-end handoff for FA/EAM

Working, audited code — not another Figma frame to interpret.

For the FA / EAM Dev Teams · UI/UX Team

From a Figma frame to working code

! The old way — Figma handoff

- A static picture — you rebuild it from scratch
- Re-derive every component & pattern by hand
- Constant context-switch: design tool ↔ code
- Ambiguity → back-and-forth to confirm intent

✓ The new way — UI sandbox

- A real, running Angular screen you can click
- Components & patterns already assembled correctly
- Copy the finished front end into FA-Suite
- Swap the mock data for the real API + logic

What a UI sandbox is

A fully-built Angular implementation of a screen, wired to mock data, living on its own branch in one repo.

Built

Finished front end

Layout, components, states & interactions — all assembled to spec.

Mock

Wired to mock data

Loaded the same way a real API would be, so it swaps in cleanly.

A11y

Light + dark, responsive

Both themes checked; behaves at real screen sizes.

1:1

One branch per ticket

Each sandbox is a single Jira ticket. main is just the shell.

The shift for you

You don't rebuild the UI.

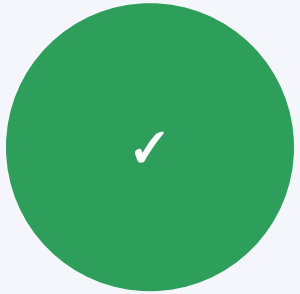
You copy it in and swap the data.

1 Copy the front end

2 Swap mock → real API

3 Wire logic & ship

The design is already signed off



Every sandbox is reviewed and approved by Product before it reaches you.

Layout, interactions, states, behaviour — those decisions are made. As the developer wiring in the logic, you don't need to reopen or second-guess the design. Your focus is the data and the logic.

And it's already audited — to production's bar

A11y

Accessibility (WCAG AA)

Contrast in light & dark, ARIA/labels, keyboard, focus.

FA

FA coding standards

Bootstrap-first, tokens, modern Angular, no dead code.

UX

UX content styling

Correct typography classes, color tokens, link styles.

CCL

Storybook / CCL components

Verified against the library's reference components.

VO

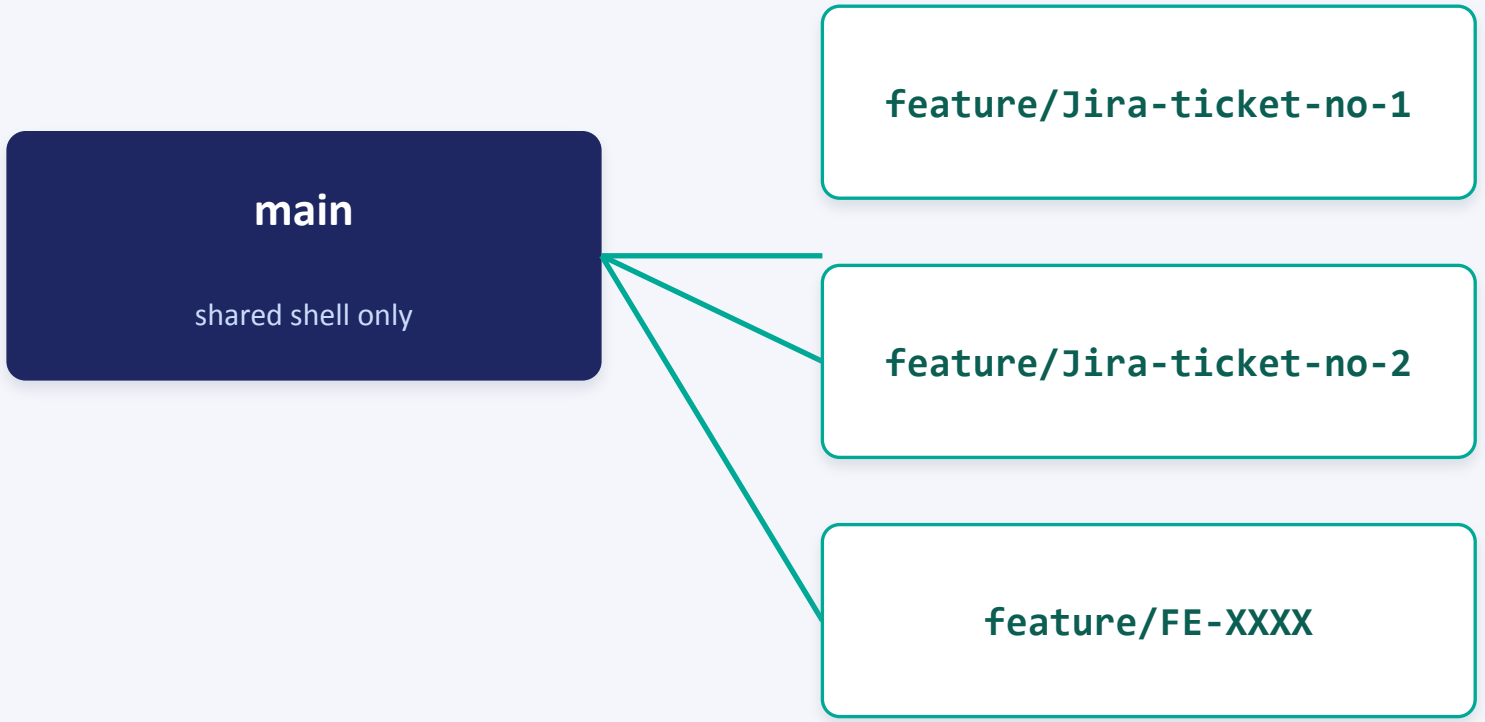
VoiceOver checklist

Manual screen-reader pass (strong, not yet exhaustive).

Committed audit report

Every branch ships one — findings by severity + an overall Pass / Conditional / Fail.

One repo, one branch per ticket



Start from the UX item in Jira

The UX item carries:

- the link to its exact branch
- a video walkthrough of the design

Don't browse the repo root — the branch link comes to you.

Your workflow: 5 steps

1. **Find the UX item** Under the epic, or via the 'Depends on' link on your dev ticket. [this part has been trickey!]
2. **Watch the walkthrough** The video on the UX item explains behaviour & edge cases.
3. **Check out the branch** Clone it, open alongside FA-Suite, run against mock data.
4. **Copy the front end into Production branch** Bring the finished UI into your Production working branch. - A claude skill has been created to help with this in the github skills repo
5. **Swap mock → real API** In Production code, wire in the real endpoints and logic. Done.

Use it in your workspace — let your AI do the work

Add the sandbox to your FA-Suite workspace so your AI sees both — it copies the front end across into FA-Suite and answers your questions from the sandbox code.

1. **Clone the sandbox** `git clone https://github.com/sandbox-repo`
2. **Check out the ticket branch** `git checkout feature/jira-ticket-no-XX`
3. **Open it alongside Production branch in VS Code** *your AI sees the sandbox (source) + FA-Suite (destination)*

Then ask your AI — instead of asking in chat:

“Copy this component into Production branch” “How does the status badge work?” “Where’s the mock data and how do I swap it for the API?”

If an API doesn't fit the design

Designs are built against what already exists — the codebase is crawled for available APIs first, so the front end should map onto real endpoints.

Hit a real conflict?

An endpoint won't do what the design needs.

Flag your Product owner

Don't force it. Don't silently deviate.

a Product loops in UI/UX for a redesign

b The three of us decide it verbally — you move on

No need to force it. Or silently deviate.

Example: PM Checklist table (FE-4000)

What the sandbox demonstrates

- A real, interactive data table (AwTable)
- Status badges derived from row data
- Filters, row indicators, status menus
- Dialogs (unsaved-changes, alerts) & toasts
- Light + dark mode, accessibility built in
- All on mock data — ready to wire to the API

Where to find it

Epic Jira ticket

Find child UX-item ticket

Sandbox branch is also linked in description of UX-item

Video walk through is also linked in UX-item description

Where everything lives

Full guide (Confluence)

UI sandbox confluence pag has been written

The repo

Within the AssetWorks Github lives the UI sandbox repo

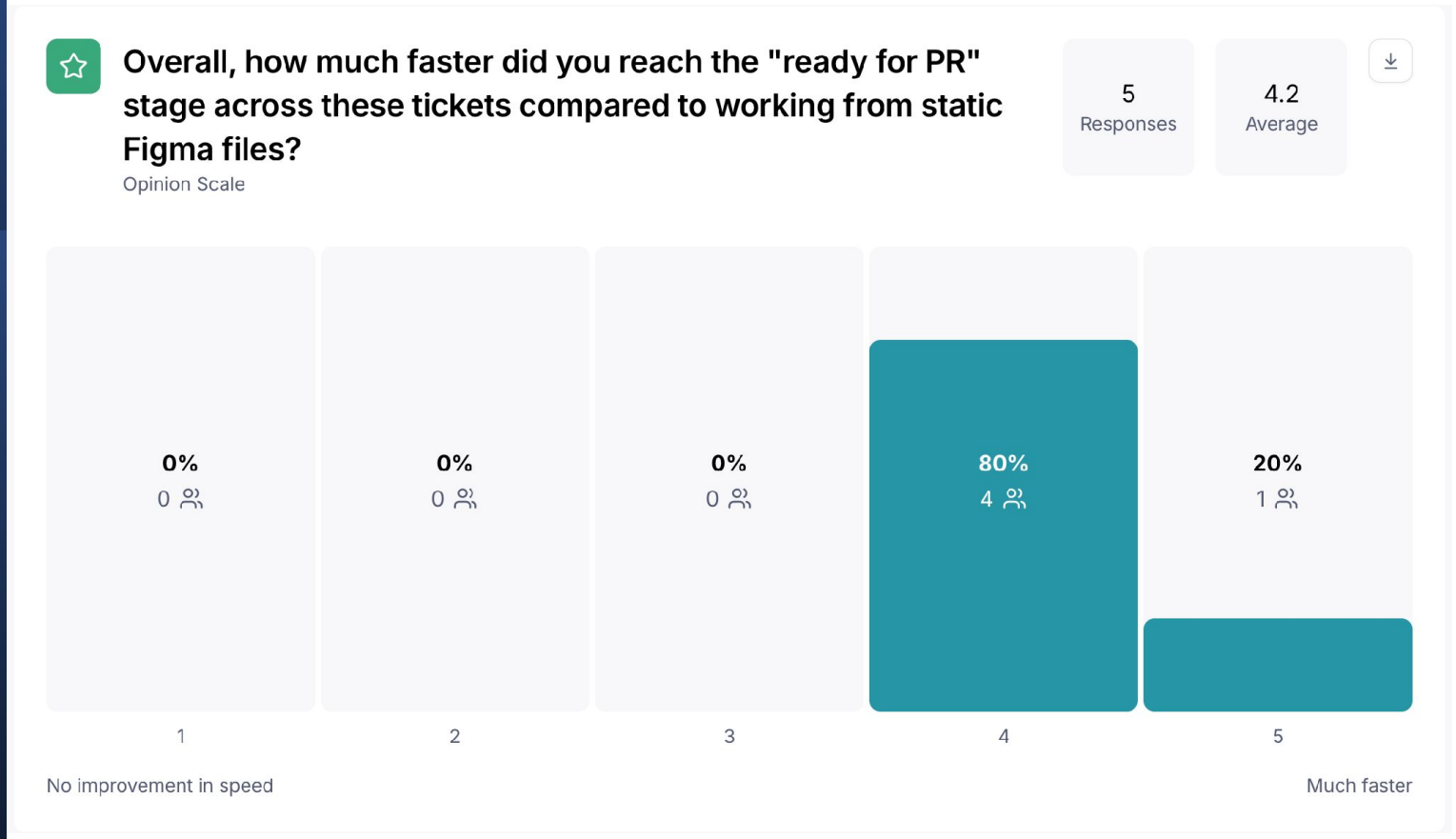
Need a sandbox or hit a snag?

Reach out to Product, they will loop me in if needed. For a sandbox tutorial, reach out to me directly.

Less interpreting. More building.

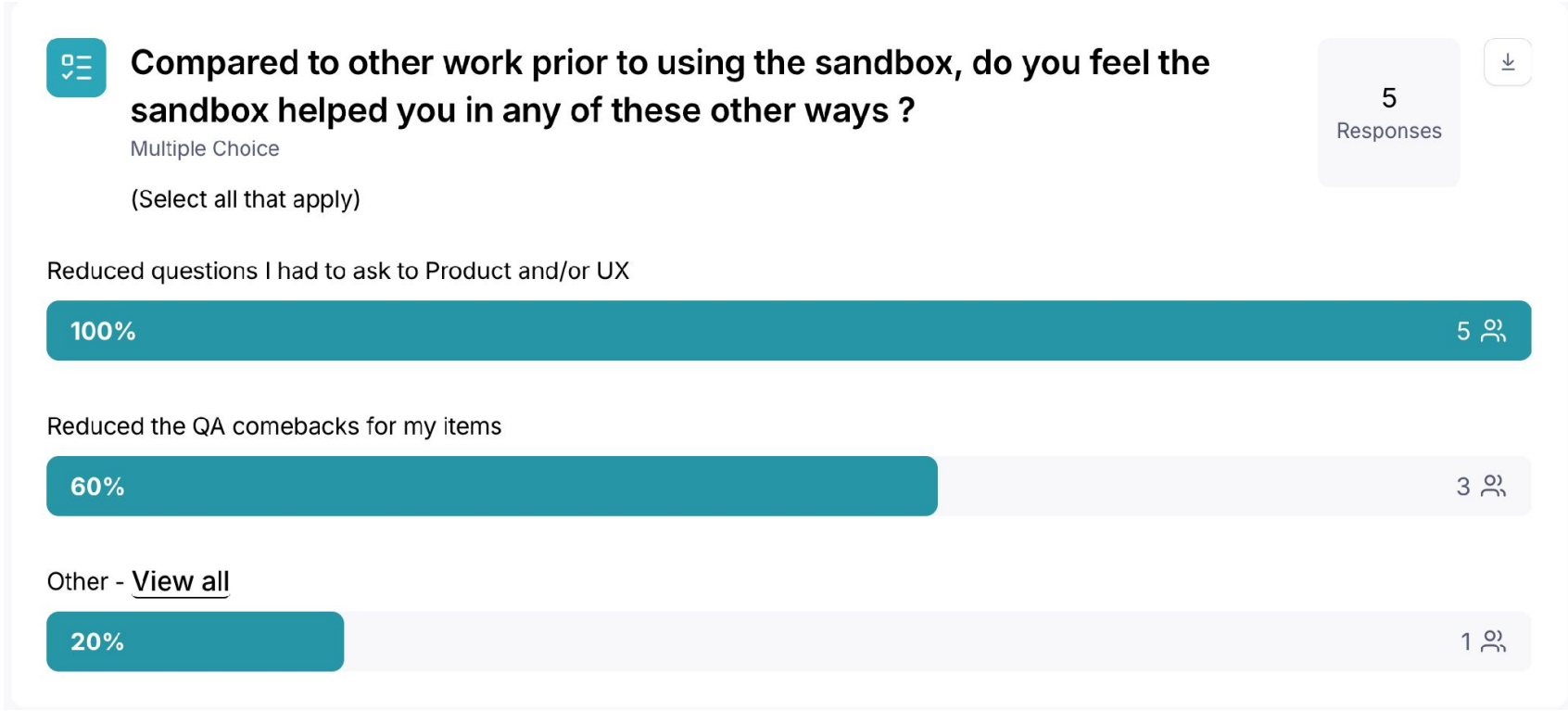
Less interpreting. More building.

Survey says...



Less interpreting. More building.

Survey says...



UI SANDBOX PILOT · 5 DEVELOPERS · 13 TICKETS · ALL FIGURES SELF-REPORTED

Developers estimate the sandbox cut their effort roughly in half

90 → 42

story points across 13 tickets
estimated without a sandbox → actual with one

53% fewer points
2.1× the throughput for the same estimate

5 of 5 devs reached "ready for PR" faster
4.2 / 5 average · zero ratings below 4

SELF-REPORTED — EACH DEVELOPER'S OWN ESTIMATE AND RECALL

Developer	Tickets	Without sandbox	Actual	Δ
P1	5	20	10	−10
P2	1	3	2	−1
P3	1	2	2	0
P4	1	50	8	−42
P5	5	15	20	+5
Total	**13**	**90**	**42**	**−48**

The "without sandbox" column is a judgment, not a Jira record — devs were asked what they'd have estimated had no sandbox existed. One ticket [redacted]) accounts for 42 of the 48 points saved.

WHAT ELSE CHANGED

It's not just story points — and it's easier than v1

100%

of devs said the sandbox reduced the questions they had to ask Product and UX

60%

reported fewer QA comebacks on the items they built

10 of 13

tickets were covered by just two sandboxes — build once, reuse across the epic

Adoption friction fell after the v1 fixes

"How difficult was it to incorporate the sandbox into your workflow?"

1 = "easy peasy" · 5 = "so hard" · average rating

UNPROMPTED — AND NEW

"Easier to send AI into the code and have it match the design. Especially helpful when something didn't look right."

The sandbox is machine-readable design context. A Figma file isn't.

Source: Maze study 516116638 · April–August 2026 · 5 developers, 13 tickets

FAQ

"What about dependencies / package versions? Won't the sandbox's versions fight ours?"
Copy the component code, not the sandbox's package.json or lockfile. FA-Suite resolves its own dependency tree. The sandbox's main is kept aligned with FA-Suite production versions, so drift is minimal — but the lockfile has no path into production.

"What if the API doesn't do what the design assumes?"
The design is built against APIs that already exist (the codebase is crawled first), so it should map onto real endpoints. If you hit a genuine conflict: don't force it, don't silently deviate — flag your Product owner. Either UI/UX redesigns for the constraint, or the three of us make a quick verbal call so you keep moving. (Slide 9.)

"Are there tests? Can I trust it?"
The sandbox is a UX harness — no unit-test runner in it by design; unit tests live in FA-Suite (Vitest). What it does carry is a committed pre-ship audit report: WCAG AA, coding standards,

UX styling, component checks, and a VoiceOver checklist, with an overall Pass / Conditional / Fail. You're inheriting vetted code.

"How 'done' is the accessibility, really?"
Contrast (light + dark), ARIA/labels, keyboard, and focus are audited and fixed before handoff. VoiceOver is a manual pass — strong, but not yet exhaustive. So: a big head start, not a blank check. Preserve it as you integrate.

"Why not just keep using Figma?"
Figma hands you a static picture you rebuild and reinterpret. The sandbox hands you working, patterned, audited code — less context-switching, less ambiguity, less back-and-forth. You spend your time on data and logic, not on reconstructing the UI.

"The branch didn't run for me."
It needs Azure DevOps credentials in .npmrc for the @assetworks-llc private scope (same as any FA front-end work). If it still won't run, the walkthrough video on the UX item shows the intended behaviour — and you can still copy the code.

"Do I have to match the sandbox exactly?"
The design decisions are Product-approved, so yes on look and behaviour. Implementation details that don't change the UX are yours. When in doubt on a UX call, ask — don't guess.

"Where's the source of truth if the guide and a branch disagree?"
The UX item + its branch are the current design. The Confluence guide is the process reference. For coding standards, the shared fa-workflow standards win.