

AI for Architects: What's Possible, and What Isn't Yet

An honest field guide, with paste-ready prompts, from evanparra.ai

If producing blueprints is a bottleneck, you have probably wondered whether AI can help. The honest answer is yes for part of the work and no for another part, and the line between them matters. This is a plain look at where AI genuinely helps an architect today, where it does not, and a set of prompts you can paste in and test on your own real work instead of trusting a canned demo.

The honest picture

AI is genuinely useful for the **front end** of design. It is not a replacement for your architect, and it does not produce final drawings. The line is simple and worth stating plainly:

- **What AI does well today:** read an existing plan and summarize it, research zoning and building code, generate early concept and layout options in seconds, and turn a plan into a presentation-ready concept image. It compresses the slow exploration work that happens before drafting.
- **What AI does not do:** produce stamped, permit-ready construction documents. Those are dimensioned, code-compliant, legal instruments that a licensed architect designs, seals, and takes responsibility for. That does not change with a better AI. The architect stays the architect.

The right way to think about it: AI is a very fast sketchpad that lets an architect explore more options in less time and spend more hours designing and fewer hours assembling. The final blueprint is still the architect's work.

The one setup fact that determines the test

Two AI systems do different jobs, and knowing which is which saves you a wasted afternoon:

- **Reasoning over the plan** (read it, extract the program, flag issues, structure a prompt): a strong text model like Claude (Opus) or Gemini. It replies in words, not pictures.
- **Image in, image out** (return a concept image or layout): an image model like Google's Gemini image model. It is the one that actually returns a picture.

The likely winning shape is the text model as the brain that reads the plan and writes a precise prompt, feeding the image model as the renderer. Run both and see.

How to test it on your own work

Do not trust a polished demo. Sit down with two or three of your own real plans at different stages, a stopwatch, and the prompts below. For each test, rate usefulness and accuracy from 1 to 5 and compare the time against the way you do it today. If a task does not test well, drop it and focus on the ones that do.

A few things to set up first: use enterprise or no-train settings before you upload any proprietary drawing, and confirm you are comfortable with that. Expect the image model to **drift** (redraw the whole plan when you wanted one wall moved) and to **invent dimensions**. Its output is a concept, not to scale and not permit grade. Image-model output also carries a watermark, fine for internal exploration, worth noting for anything client facing.

Paste-ready prompts

Copy these as written and attach your blueprint image. Bracketed text is what you swap per drawing.

Test 1: Comprehension (use the text model first)

I am attaching an architectural floor plan. Do four things, and be specific about what on the drawing you are basing each point on. 1. Program: list every room or space you can identify, with its label and approximate square footage if shown, and give a total. 2. Circulation and adjacencies: describe how a person moves through the space and which rooms connect to which. 3. Concerns: flag anything that looks like a potential building-code, egress, accessibility (ADA), or clearance issue, and explain why. 4. Unknowns: list what you cannot determine from this drawing and would need to confirm. Do not guess at any dimension that is not shown on the plan.

Watch for: does it read the plan correctly, and are the flagged concerns real? This test needs no image output, so run it in the stronger text model first. If a specific code or accessibility question comes up, do not let the model rule from memory: paste in the relevant public section of your local code or ADA standards and ask it to reason against that source and cite it. Treat the answer as "here is what I found, verify against the adopted edition," not a ruling.

Test 2: Localized edit (image model)

Here is a floor plan. Produce a top-down 2D floor-plan image that is identical to this one except for one change: [move the reception desk to the north corridor]. Keep the overall footprint, the exterior walls, every other room, and all room labels as close to the original as possible. Change only what I asked for.

Watch for: drift. Does it keep the rest of the plan or redraw everything, and does scale hold? Try two or three single edits (add an exit, merge two offices, widen a corridor) and see if any stay coherent.

Test 3: Option generation (image model)

Here is the footprint and interior program of a [office / clinic / retail] floor plan. Generate 4 alternative interior layout options for this same footprint, each as a labeled top-down 2D floor plan, optimized for [flow and efficiency]. Keep the exterior walls and total area fixed and vary only the interior layout. Briefly label what is different about each option.

Watch for: this is often the strongest use. Are the options plausible and usable as thinking starters? Time it against how long four hand-sketched options normally take.

Test 4: Visualization (image model)

Here is a 2D floor plan. Produce a realistic 3D concept rendering of this space as built, suitable for a client or investor presentation. Give me [an exterior massing view / an interior view of the main entrance and lobby]. Keep the layout and proportions faithful to the plan.

Watch for: judge it as a presentation visual, not a measured drawing. Note any visible watermark before using it with a client.

Test 5: Reasoning plus render loop (text model, then image model)

Step A, in the text model (attach the plan):

Based on the attached floor plan and your analysis of it, write a precise, detailed image-generation prompt I can hand to an image model to produce [the requested change or option]. Specify the room layout, adjacencies, any known dimensions, the viewpoint (top-down 2D or a named 3D view), and the style. Make it specific enough to reduce the chance the image model invents details.

Step B: paste that prompt into the image model and attach the original plan.

Watch for: does the two-model loop (text model structures, image model draws) beat prompting the image model directly? Compare this output to Test 2 or 3 on the same change.

What success looks like, and where it could go

If your own testing shows real time savings, the next step is a small pilot: use the tools on live work for a week or two, and measure the result. If that holds up, the natural next step is a simple agent, running on your own cloud, that you use in plain language, with every output reviewed before it is used. The most useful version of that agent likely reads a plan and checks it against the building codes and accessibility rules for that project's type and location, pointing to the specific rule it relied on so you can verify it quickly. It flags and cites; the architect still owns the final call.

The principle holds the whole way through: AI gives you faster exploration and a second set of eyes. It does not give you a signature. Keep that line clear and the rest is upside.

About evanparra.ai. Evan Parra builds practical AI and custom software for technical businesses, on Google Cloud. If you want a second set of eyes on how AI could fit your operations, or on a piece of software you are considering buying, reach out at evanparra.ai. No pitch, just a straight answer.