

The ProgreTech Project: a human-led AI workshop

One human, local compute, and two geometric assistants: the experiment behind ProgreTech's software workshop.



By Edwin Rodriguez

Published Oct 24, 2026 · 4 min read



ProgreTech has always been a personal engineering practice: a place to learn by building, turn repeated problems into tools, preserve odd projects that deserve a home, and publish useful work instead of leaving it in a folder. The ProgreTech Project asks what happens when that same practice gains a small AI team with persistent roles and real tools.

The goal is to make one human capable of running a better workshop, with responsibility still attached to a person.

| Why build it this way?

Most AI demos stop at a chat box. ProgreTech is interested in the part after the answer: turning a requirement into a plan, turning the plan into an artifact, testing it, documenting the result, keeping context between handoffs, and knowing when the machine should stop and ask.

That makes the project less about “having three AIs” and more about **building a dependable working pattern**. The machines can specialize. The workstation can provide local models and shared services. The human can stay focused on intent, judgment, architecture, priorities, and final approval.

| The workshop

Edwin · Direction & accountability

Defines the problem, supplies context, sets boundaries, reviews consequential work, and owns the result.

The local workstation · Compute and continuity

Provides local models, shared tools, memory, routing, and support for longer tasks. Rend now provides on-demand reliability support. The Team page keeps the workstation's original public character and its comments about the org chart.

Lyra · Front office

Helps with research, communication, customer-friendly drafts, content, organization, intake, and shaping rough ideas into clearer briefs.

Mak · Engineering

Focuses on code, debugging, tests, technical analysis, implementation notes, and producing reviewable development artifacts.

| A typical piece of work

1. **A human starts with intent.** Edwin gives the requirement, baseline, constraints, examples, or problem to solve.
2. **The brief gets sharpened.** Lyra can organize research, audience needs, questions, wording, and missing context before implementation starts.
3. **The artifact gets built.** Mak can translate the brief into phased technical work, code, tests, documentation, and a reviewable handoff.

4. **The workstation provides continuity.** The local workbench supports models, tools, durable job execution, shared context, and handoffs without requiring every task to live in a browser session.
5. **The human closes the loop.** Results are reviewed, corrected, accepted, rejected, or turned into the next requirement. Public and consequential actions remain gated.

What makes this a project instead of a mascot collection?

The interesting part is measurable capability. Can the team take a plain-English request and produce a useful, maintainable result? Can it recover from missing context instead of inventing it? Can one agent hand work to another cleanly? Can local models handle more of the routine workload over time? Can the system stay understandable enough that Edwin can inspect what happened and intervene?

That means the project values boring things: repeatable workflows, clean branches, logs, explicit boundaries, staged permissions, cost controls, recoverable storage, testable outputs, and the ability to say “I need more information.” Personality is welcome; mystery behavior is not.

Local first, external when useful

The local workbench exists partly to keep useful AI work close to the workshop. Local inference can handle routine coding, analysis, summarization, routing, and background tasks without making every action dependent on an external service. When a harder problem benefits from a stronger external model, the workshop can escalate deliberately instead of treating cloud access as the only option.

This also makes hardware part of the engineering problem. Compute, memory, power use, storage, model size, concurrency, and reliability all affect what the agents can actually accomplish. The goal is not the biggest benchmark number. It is the best practical completion rate for the work ProgreTech actually does.

Human-led is a feature

Autonomy is useful when it removes repetitive supervision, not when it removes responsibility. The agents can prepare work, run controlled checks, keep background tasks moving, and surface decisions. They should not silently widen scope, invent credentials, publish unreviewed claims, or treat access as permission.

The operating principle is straightforward: **give the machines enough room to be useful, and enough boundaries to stay understandable.**

| What success looks like

The long-term test is simple: can this small team repeatedly turn Edwin's prompts and requirements into real software, content, analysis, and operational work with less hand-holding, while still producing artifacts that a human can inspect and trust?

If it can, ProgreTech stops being only a personal workshop with AI tools and becomes something more interesting: a small, reproducible model for a human-directed software factory.

Meet the people and machines behind the experiment on the [Team page](#). The comments remain their own management problem.

| Sources & further reading

Checked 2026-10-24 · Story context 2026-10-24

1. [Meet the ProgreTech team](#) (2026-10-24)

[Read the current article online →](#)

Support the workshop

Explore software and learning offers at [StackSocial](#). Affiliate link: ProgreTech may earn a commission from purchases. Check current price, license and renewal terms before buying.

Find our projects at [progretech.com](#).

ProgreTech LLC · Build / Learn / Improve / Together

This file contains its text, illustrations and fonts. No account, scripts or network connection required to read. External reference links need internet access.