

A₂D

AI Augmented **D**evelopment

AI Assisted **D**evelopment

A2D is not vibe coding

Vibe Coding — Definition

Vibe coding is a development approach where the human sets the *direction* and the AI handles the *details*. The emphasis is on speed, exploration, and trusting the AI to fill in the gaps. Developers rely heavily on natural-language prompts, minimal technical scaffolding, and fast feedback loops—often generating entire features or projects with loose guidance.

Key Characteristics

1. High-level prompts, minimal specificity

You describe *what you want*, not *how it should be built*.

“Give me a full-page login screen with Google OAuth and dark mode.”

2. AI handles most of the implementation

The AI writes all or nearly all code, tests, docs, and even architecture.

3. Fast iteration and regeneration

If something doesn't look right, you regenerate or adjust the prompt rather than manually refactor.

4. Low-touch validation

You may skim code but don't deeply review or reason about it in the same way as traditional development.


5. Creative exploration

Good for brainstorming, rapidly prototyping, or generating UI/UX variations.



Dream it. See it. Ship it.

Transform ideas into full-stack intelligent apps in a snap.
Publish with a click.

 Spark is currently only available to Copilot Pro+ and Enterprise users. [Upgrade now to access Spark.](#)

Create web apps with React and TypeScript to prototype ideas, build tools, and more

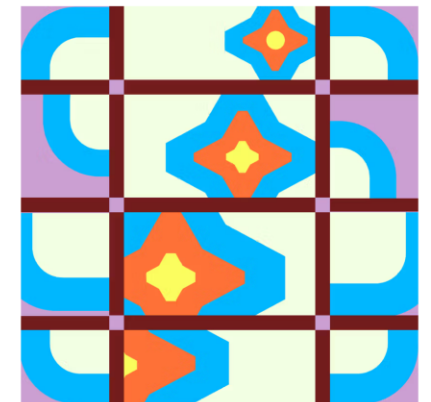


[Spark](#) uses AI. Check for mistakes. [Terms of service.](#)

Figma AI is your creative collaborator

From prompting prototypes to speeding up workflows, Figma AI helps teams bring their best ideas to life.

[Learn about our approach to AI](#)



A2D is
not vibe
coding

The AI Development Spectrum



Traditional Development

Developers using little to no AI in their workflow



Vibe Coding

Teams fully embracing AI to build and deploy production software



A2D (AI Augmented Development)

A2D optimizes human-AI collaboration.

It redefines how software is planned, designed, built. A2D uses intelligent agents that become trusted members of the development team.

Demo Time

Agents

Delegate tasks to GitHub Copilot coding agent to work on in the background, and then monitor Copilot's progress.

Describe a coding task to work on

 Dave76/tlwl-cubo ▾



Recent sessions

No sessions found. Create one by se

✓ None

codelantern-architect

codelantern-coder

codelantern-planner

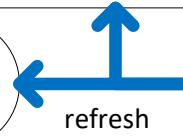
+ Create an agent

Open 0 Closed 9

A2D Methodology

A2D Workflows

A2D Artifacts

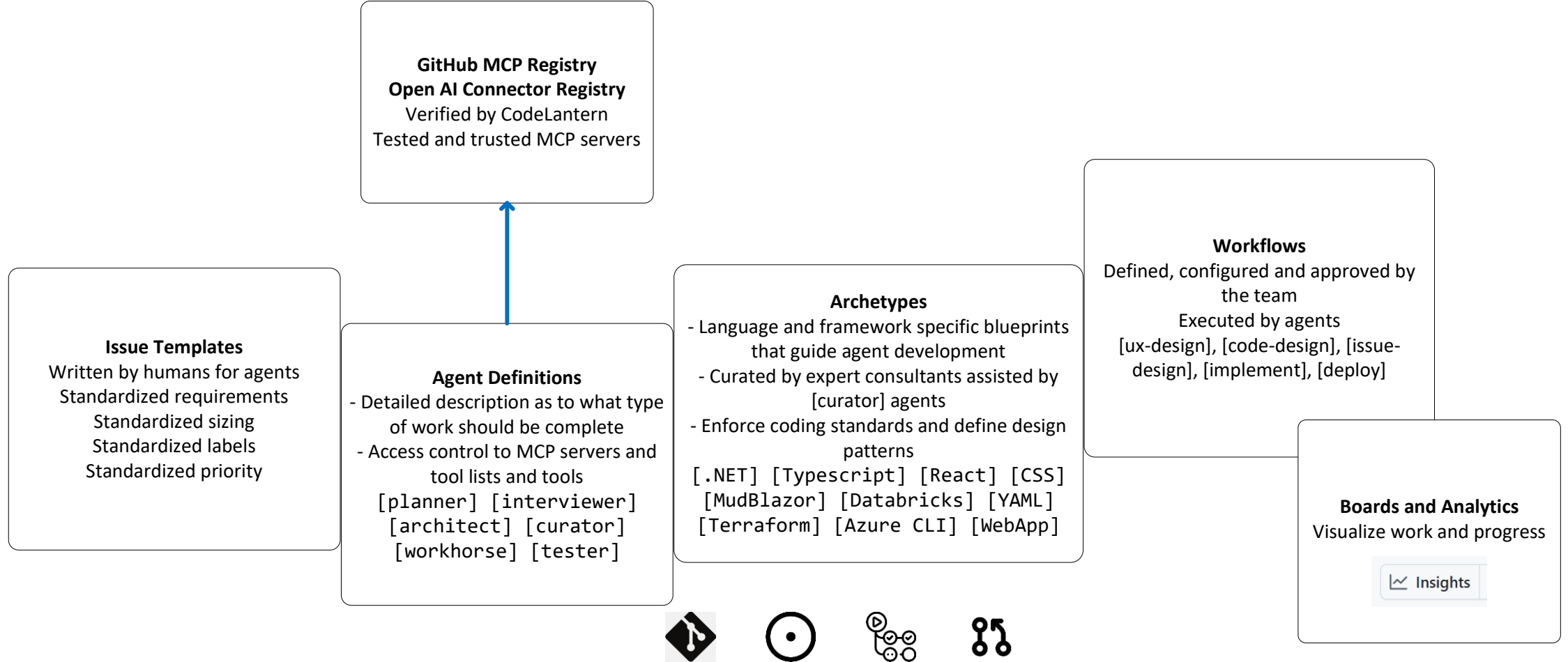


A2D Subscription

GitHub

GitHub Copilot

Open AI, Anthropic, Gemini



GitHub, GitHub CoPilot, and GitHub Coding Agent form the foundation for A2D. Agent orchestration, tool permissions, workflow, source control, pull requests, work items, projects, labels, releases. As of now, the Coding Agent uses Claude Sonnet 4.5



GitHub Copilot



OpenAI

AI



Gemini

Component Definition

Artifact	Description
Work Item Templates	Issues, Sub-Issues, Pull requests all must structured in a consistent form. Agent definitions will reference these templates for both read/write operations.
Agent Definitions	Describes agent role and purpose. Configures access to “tools” and to enable “workflows”. Reinforce use of archetypes and expectations of output.
Archetypes	Detailed and structured documentation about the frameworks and patterns the agents should use. Curated by experts with AI assistance.
Workflows	Defines the actual work to be completed, who participates (AI and Human), with a clear concise goal or outcome.

Hiding the IP

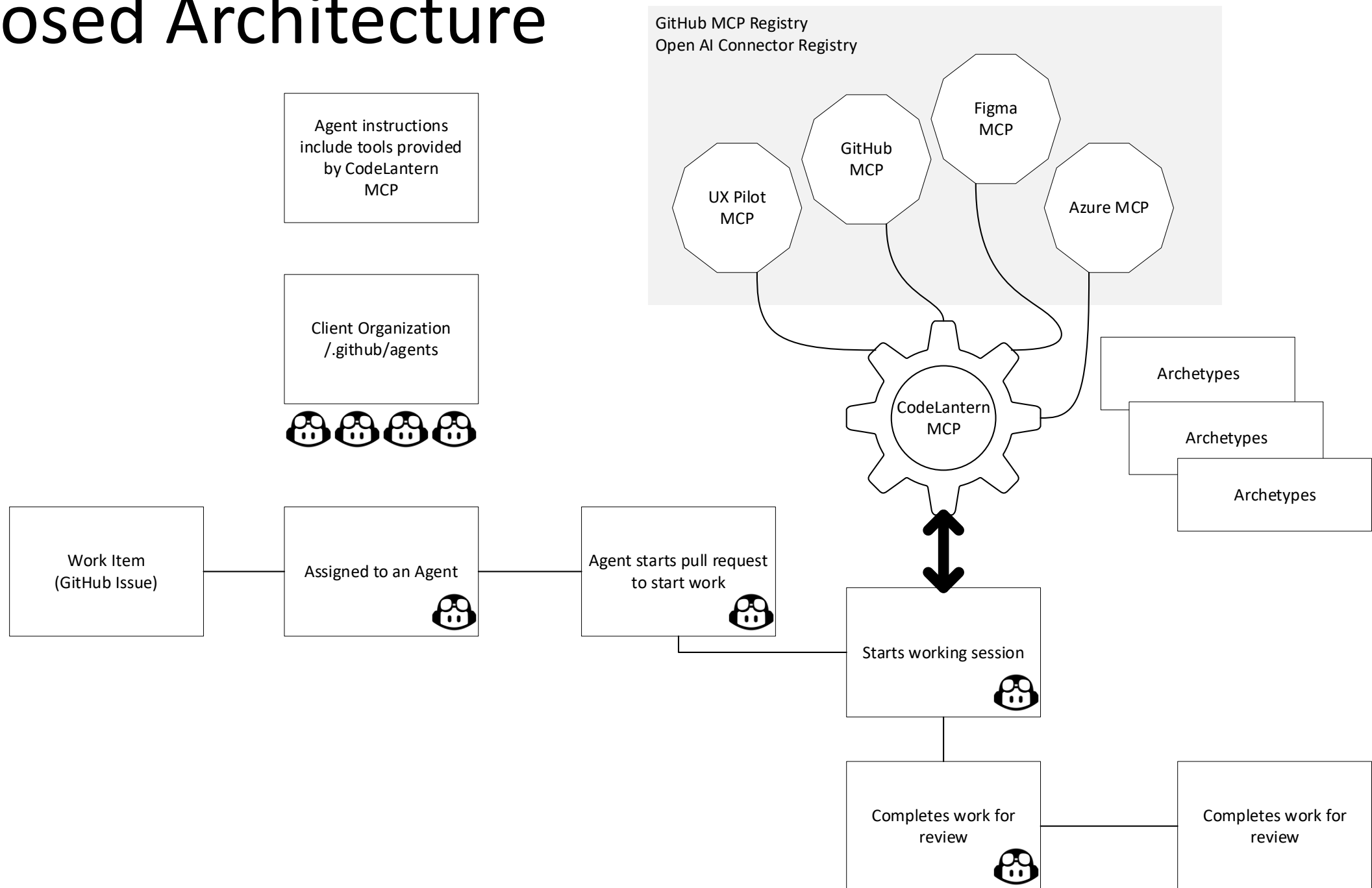
- Since agents rely on definitions, archetypes, and workflows, these components are produced in markdown format to ensure they remain easily readable.
- This format presents challenges in concealing, updating, and managing the artifacts as proprietary, as well as maintaining synchronization with a central repository.

Conceptual Proposed Solution*:

- Safeguard intellectual property through a dedicated MCP server.
- Deploy the MCP server via the GitHub registry.
- Issue access tokens to paying customers and subscribers for MCP server access.
- Implement support for multiple versions to maintain backward compatibility and accommodate potential changes in the MCP environment.

**Currently theoretical and unconfirmed*

Proposed Architecture





Workflows



Planning Workflow

- The planning workflow uses the [planning-agent] to ensure work is well defined and has acceptance criteria.
- The [planning-agent] can ask questions, update work items, and split large items into multiple sub-items and tasks.
- It can recommend next steps by analyzing what needs to be completed. IE: understanding the task is to create a user interface and recommending the [UI Design Workflow]
- A planning conversation is a quick way to build up the backlog.



Suggestion Workflow

- Scan the current set of issues and make a simple work item describing what the team should work on next.
- Used by product owner and architect
- Uses [planning-agent]
- Ability to create work items on the board with thoughtful next steps
- It can look at comments, markdown file checklists and gaps in coverage to suggest what should be worked on next.

User Interface Design Workflow (future)

- Describe the interface that needs to be built
- Generate potential UI options using archetypes defined for the current project (Material Design, MudBlazor, Tailwind)
- Present and Refine
- Attach to new work item for implementation





Solution Design and Analysis Workflow

- Deep analysis looking through archetypes, current repository, and similar problem statements
- Presents back 2-3 options with detailed approach, code samples, and documentation in markdown format
- Score the options and make a recommendation
- Also provide insight as to whether this should be done now or can wait until later
- Examples:
 - Should we refactor the app to use a shared caching model?
 - Should we implement the bUnit testing framework?
 - Should we refactor the frontend to use React instead of Blazor?

Implement Workflow

This is the real work horse of the system and “pumps” out the code and tests.

The [coder] agent is given instructions to:

- Be concise and efficient with coding changes and to minimize refactoring (unless explicitly instructed to do so).
- Read and understand existing design patterns and architecture decisions.
- Write appropriate unit tests.
- Write well documented pull requests with screenshots where it makes sense.



Deployment Workflow

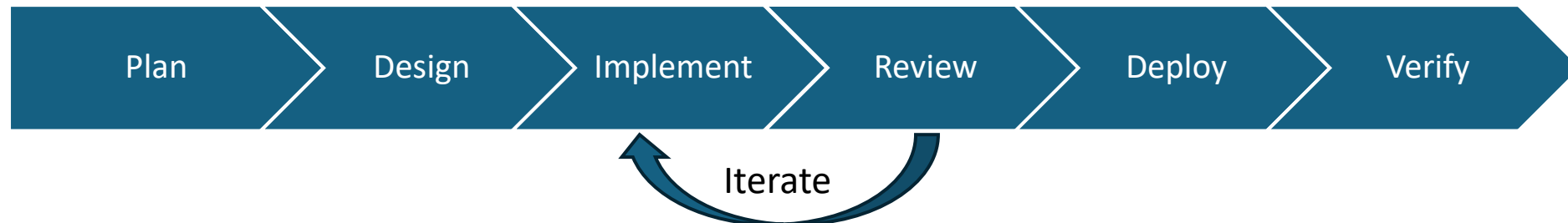
- Uses [deployment] agent who will have expertise and knowledge of Terraform, Azure CLI, YAML, GitHub actions.
- Inspect solution to ensure components support automated deployment
- Create terraform and scripts for deployment
- Ensure environment consistency
- Ensure secrets remain secret
- Ensure code can be deployed to an environment

Workflow Status (not all a required for mvp)

Workflow	Status	Notes
Planning	In Progress 75%	Process chat commands and/or inspect a work-item, enhance description, acceptance criteria, and can split work-item into multiple parts. <i>Needs to be less verbose. Needs to ensure process allows for PR approval prior to issue modification/creation.</i>
Solution Design	In Progress 80%	Process chat commands and/or inspect a work-item, analyze the repository and archetypes, then compare/contrast multiple paths forward – and make a final recommendation. Creates detailed pull requests with markdown documents. It is too verbose and needs tuning.
Suggestion	In Progress 60%	Working prototype. This agent can help decide what to work on next, which could be an issue on the board, or something that needs refactoring or attention in the code base. Example inspecting TODO comments, markdown checklists, and lack of code coverage from a unit testing perspective.
User Interface Design	Planning 25%	NOT WORKING: When a work item requires a visual mock-up, screenshot, design it can work with third-party design MCP (Figma/UX Pilot), to generate screenshots. Much better than relying on the [coding-agent] to figure this out during the coding process.
Implementation	In Progress 90%	This is the most well developed and testing. With proper instructions and archetypes, it can use the stock sonnet 4.5 coding agent (or codex pro agent) to produce concise, clean, and tested code. Pull requests are excellent. Documentation is excellent. Lack of code drift and wander is also excellent. Approval gates are well defined.
Deployment	Planning 25%	The coding agent can already build excellent deployment scripts and infrastructure as code, but lacks understanding of process/actions/workflows to tie this into the GitHub ecosystem. Needs strong approval gates.

Methodology Definition

- Work in progress, and perhaps more related to the service side of things, but the process is likely as important as the underlying technology – like the ceremonies in agile.
- A2D is going to require ceremonies, but with quicker iteration
- More suited to a kanban style of working because of the velocity of work completed.
- *The methodology I have been following is daily planning in the morning, followed by review before lunch, resolving comments and iteration, deployment and testing in the afternoon. Repeat.*



The human role

- Provide context to the agents
- Provide positive and negative feedback to completed work
- Review and approve all changes
- Set direction and priorities



Risks

- GitHub Copilot is advancing quickly, with numerous features still in preview and no assurance they will be officially released soon. This space is also changing at an incredible pace (likely due to the use of AI augmented development). GitHub just release Agent HQ, and Sparks v2, Figma has a tool, Cursor, Claude Code.
- The MCP registry is currently unstable. I am working with multiple MCP servers from this registry, and agents often report bugs, which means CodeLantern is being developed in a constantly evolving environment.
- GitHub Copilot maintains complete control; for example, it limits coding agents to work exclusively with Anthropic Sonnet, including any custom agents you develop.
- The cost of GitHub Copilot licenses ranges between \$15 and \$30 USD per user each month, covering the main features. There is no way the costs can stay this low. I am paying \$30 per month for 300 premium sessions, which translates into 125-150 completed work items per month. My guess is that this is going to increase significantly which may cause customers to shy away from the technology.
- Agent Drift – This is real. You can give the same context, instructions, archetypes, at the exact same spot in the repository and will produce results with variation.



Phased Approach?

- Start as a service company until enough artifacts are battle tested that they can be packaged as subscription service
- This will also allow MCP servers to “settle” with increased reliability
- It will take time (and experts) to construct archetypes, finalize agents and workflows
- Ability to generate revenue while building/testing artifacts and the MCP server
- Once the tech is proven we can start licensing it or include it with the consulting but sell at a much higher hourly rate.
- Phase 1 – Enablement, embedded consulting
- Phase 2 – Use A2D to modernize apps and solve business problems.

What to do with the funds?

1

Secure key resources including Jeff, PVZ, Nick
Take a modest salary to focus 100% of my attention on A2D

2

Build out the key artifacts, workflows with custom MCP integrations, and confirm feasibility of building our own custom MCP server to hide IP.

3

Build demos and material to attract and train key delivery consultants.

4

Pay for software licenses, operations, and key conferences/marketing opportunities.

5

Get reference customers and start building Microsoft relationship
Sales collateral / brand / social media presence
TESTIMONIALS

Next Steps

- Work on funding
- Work on securing Jeff, Nick, Peter
- Publish working demo of MVP:
 - Planning agent
 - Architect agent
 - Coding agent
 - Deployment agent
- Document and diagram workflows that can be used in conjunction with demos to attract delivery consultants and first set of customers
- Work on Designer workflow demo (this will be real wow factor)
- Work on custom MCP to hide IP

Nick Van Exan

- He's interested in joining – sick of his day job at Centrilogic
- He's only signed until Nov 30, but there is an extension looming
- He currently makes \$125 per hour – 20+K per month.
- He is asking for 15K per month for full time work hours and to offset the difference either a bonus or some equity (or both)
- My hope is that we can get him at least part time in Jan, then full time starting Feb 1
- Budget 175K for the year

Hey Dave, just following up from Wednesday. You asked me to think about what I'd need to join the A2D effort next year.

I'm currently doing about \$20k/mo (~\$125/hr, 40 hrs/week) with Centrilogic, but I can operate on reduced earnings of ~\$15k/mo. That would be the ideal target income I'd be looking for next year.

Whether that means working fewer hours (~30hrs/week) or at a lower rate (40hrs/week at ~\$94/hr), I'm pretty flexible on that. If I'm doing full-time hours at a reduced rate, it would be nice to have some potential upside to offset that at the end of the year — whether that be some kind of performance -based bonus, small bit of equity, or similar.

Not sure if that gives you the information you need, but that's sort of the ballpark I'm thinking about.

I'm flexible on structure and happy to chat more about what makes the most sense for the first year.

1:19 p.m.

Thanks Nick. That gives me something to work with before Monday's meeting. The only other thing would be to know how long your contract is going to go with GNFL. I have a long standing relationship with Kristen, so I need to make sure that you leaving is not going to cause a major issue for her. Perhaps that something I should take up with her.

1:30 p.m. ☺

BUT, I guess that's ultimately your decision too - because if you dont want to work somewhere, then you shouldn't have to.

1:31 p.m. ☺

Anyways, until I have paper in front of me this is still just a discussion. But it would be a great way to spend next year! Thanks again and have a great weekend!

1:32 p.m. ☺

You

Thanks Nick. That gives me something to work with before Monday's meeting. The only other thing would be to know how long your contract is going to go with...

Right. Ok I haven't discussed anything with Kristen (or anyone else) yet, but here is where things stand with GNL. My current extension runs to November 25, and the next one they sent me a week ago goes to January 31. Although the team is talking about an April launch and seems to assume Dan and I will stay through then, the client has not committed to that. They are also letting Val and Vlad go this month for budget reasons, with only tentative talk about bringing them back for a later phase. Vlad could step into my role if needed.

Given that, my options look like:

- Decline the January 31 extension, which would be disruptive and likely hurt my goodwill with everyone on the GNL team. They also might say that I still owe them 45 days notice because this is an active engagement (it's a bit... [Read more](#))

You

Anyways, until I have paper in front of me this is still just a discussion. But it would be a great way to spend next year! Thanks again and have a great weekend!

Thanks, Dave, you too!

2:05 p.m.

Jeff Zado

- He's very interested – called me as soon as we hung up today, but will only commit part time, which I think works in our favor.
- 20-24 hours per week.
- He didn't request a specific \$\$\$ amount – but I know Centrilogic was paying him 10k a month for 12 days of work. I know we could get him for less assuming there was upside for him. His day-to-day life expense are covered already.
- He would like any company related expenses covered: IE, lunches / dinners with clients, travel, attending GitHub universe, etc.
- More important to him was “participating” in the company success – whether that is an equity buy in at some point and/or commission percentage.
- I asked his interest in shared commission strategy with Peter so that they could drive this together, which he liked.

Peter Vander Zouwe

- He's very interested as well –he called me back with his numbers 2 hours after we hung up (not as fast as Jeff, but he needed to discuss with his wife AND his situation is not as easy as Jeff because he needs to plan an exit at Centrilogic)
- He is not happy at Centrilogic and not thrilled with their ability to execute in 2026.
- He would be really happy to be able to work with me again (and the Mikes in whatever capacity that is).
- Peter makes a current base of 8K per month ~ 100K per year. He needs to see at least that to cover his monthly expenses.
- He would be open to a shared commission model with Jeff and taking less commission if it meant being able to get equity moving forward.
- He also mentioned having a piece or stake in the company success would be greatly appreciated.

Minimize Burn

- I am going to work on this regardless of what happens from now until the Christmas break. All day every day so that we have an MVP ASAP.
- Not everyone can start Jan 1. Likely only Jeff and me. Nick and Peter will need to execute exit strategies at Centrilogic and will likely be part time until this happens. Nick's time in the beginning is more important than Peters.
- Jeff and Peter could each work 2.5-3 days per week (so together we have 1 sales rep). I have thought of using one or the other, but them working together is something special. They help motivate and teach each other. They also have different but key relationships at MS and clients. They are both open to a shared commission model. I was thinking they split 8% but would like your opinion on that.
- Jeff will take much less than 10K per month if there is upside for him. He would likely do this for no money if there was more potential upside, so something to consider. I am thinking 5K for budgeting + 4% commission.
- Peter will need the 8K per month as a base, but once commission kick in, he did say he would lower the base. Not sure if that is a positive or negative, but something to consider. 8K + 4% commission + ability to convert
- Nick will likely need the ~15K month as a base (pro-rated based on how many hours he can contribute in a given month).
- There are some key operating expenses that we cannot avoid, and we do need to budget for sales expenses and attending key conferences like GitHub Universe.
- At the end of the day – I do want the 3 of them to have the ability to share in success of the company. BUT, I agree about the buy in part, just like Shane and I did. So, perhaps the idea could be that instead of paying out commissions to Peter and Jeff, we let them convert that into equity at the end of the first year. And for Nick, instead of paying out a bonus as he asked for to get him closer to 20K per month, he can convert that into equity. And, my thought on converting commission/bonus to equity would be done at some factor (2:1 or 3:2). TO DISCUSS.