PBI CI/CD From Basic Automation to Agentic AI

Joe Rossouw

18 November 2025

Dublin fabric user group

Speaker

- Social work ☐ Epidemiology + Biostatistics ☐ Power BI

- Leading awesome team, doing interesting things

- linkedin.com/in/joe-rossouw
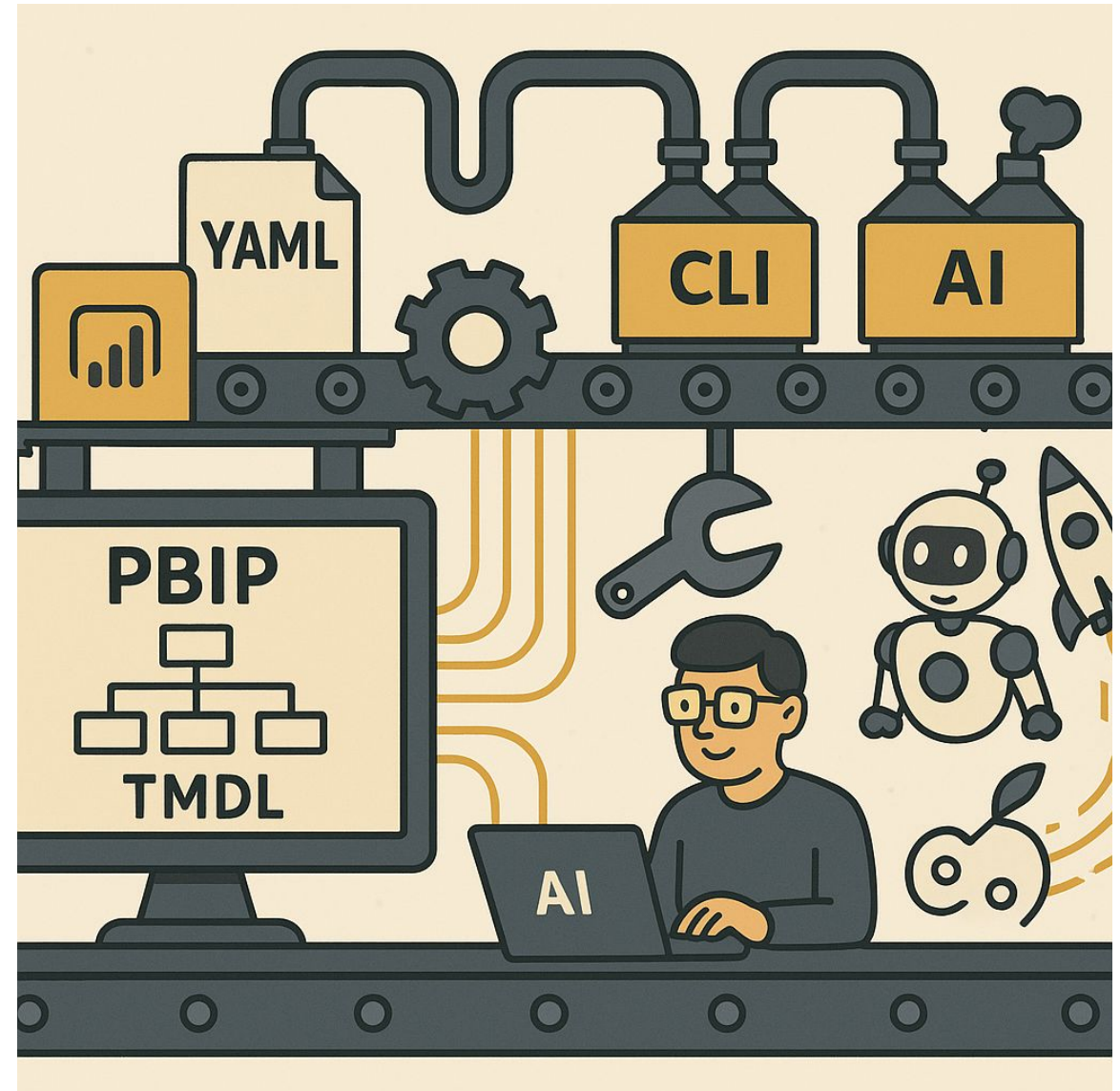
Next project:
- github.com/JoeRossouw/semantic_ops

Why CICD?

- Version control
- Collaboration
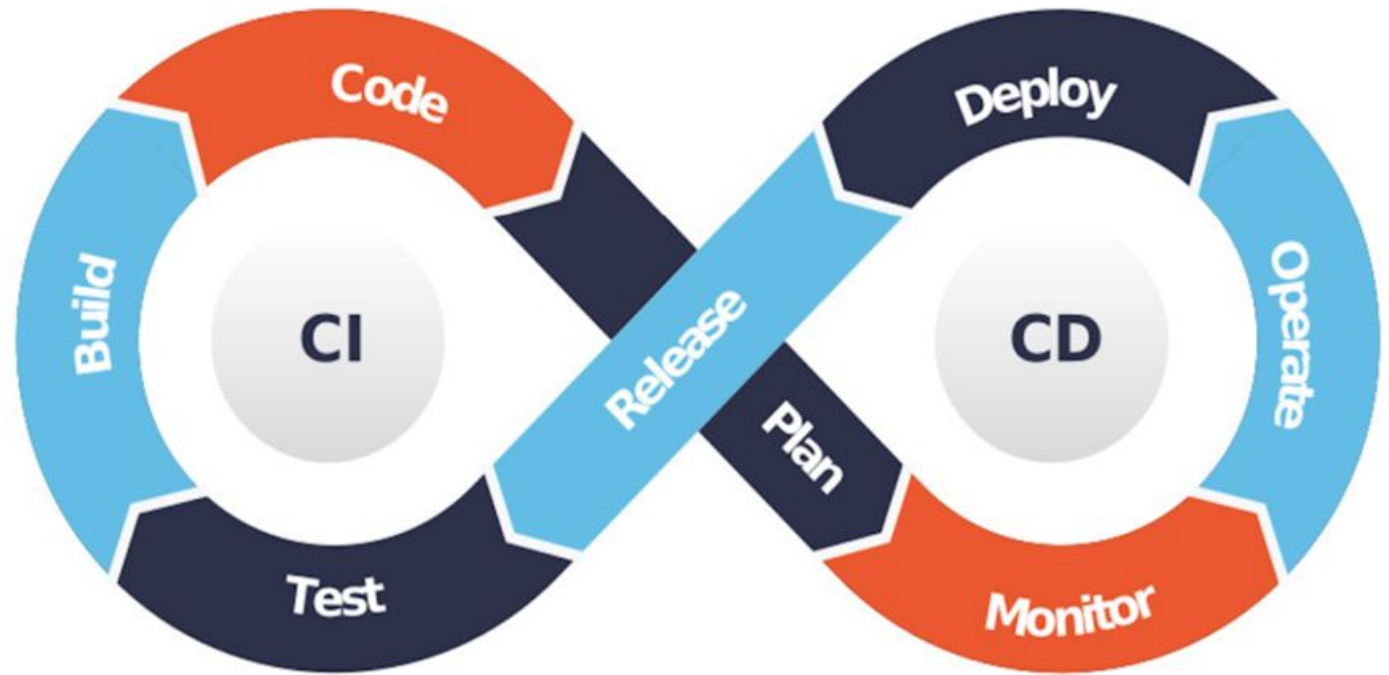- Visibility & oversight
- Quality control

- **CICD as enabler:**
- Automation
  - Deployment
  - Quality
  - Metadata
- Agentic AI
- Professional growth

Message: you cannot afford to wait!

Source control

BPA

Code review
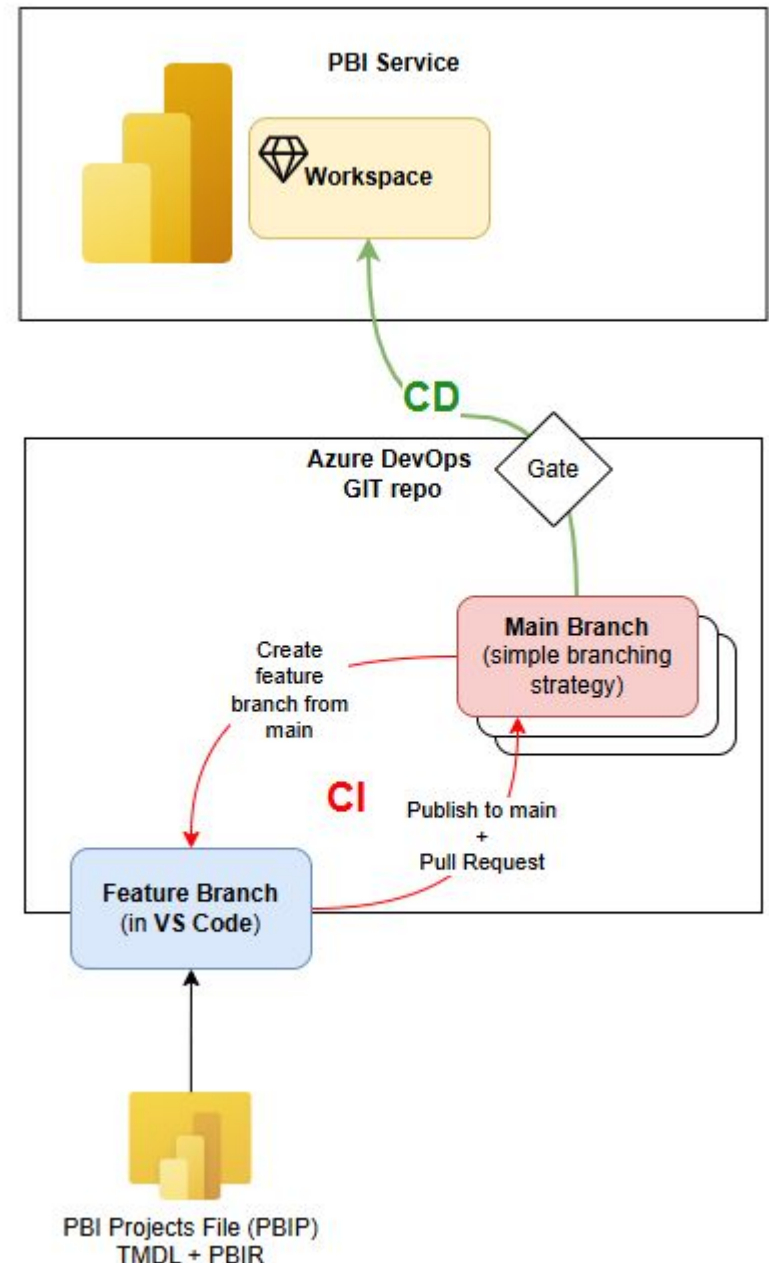
Automation

Agentic AI

Metadata

What is CICD

- **CI** = Continuous Integration
- **CD** = Continuous Development/ Deployment
- Specialized technology (e.g., VS Code, GitHub / Azure DevOps)
- Provides a systematic method for:
  - CI: integrating code to central repository, and
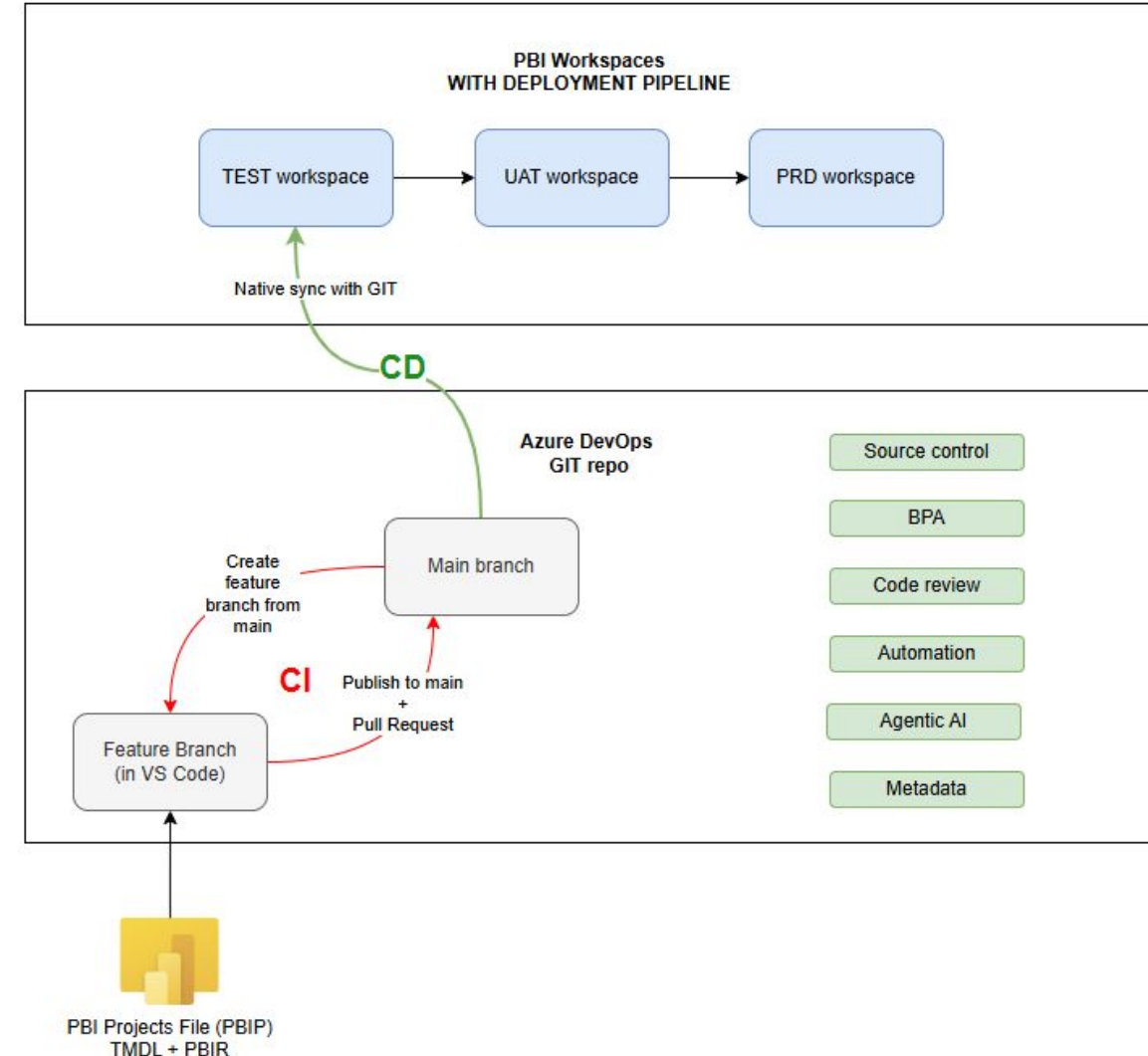 - CD: deploying solutions reliably

What is CICD in PBI

- **PBI Projects (PBIP):** Saves report + model to text files
  - Tabular Modal Definition Language **(TMDL)**
  - PBI Report **(PBIR)**
- **VS Code** (or other): Code editor - supports automation (e.g. python / PowerShell) and agentic AI
- **Git**: Technology that tracks every change + version control.
- **Repo**: A Git-managed project folder containing all files, code, and metadata.
- **GitHub / Azure DevOps**: Platforms that use Git repos. Various versions of code (**branches**). Allows for **Pull Requests** between branches to merge code. Various branching strategies possible.
- **Deploy (CD):** Native PBI git merge or CD Pipelines in Azure DevOps (or similar)
- **Gate:** Informs decision to deploy, automated or manual

Basic Setup

- Premium / Fabric Workspace

- Install VS Code

- Install Git

- Create GitHub account + repo (or Azure DevOps)

- Clone GitHub repo to local folder (accessed through Code Editor or Explorer)

- Enable GitHub Copilot (free or paid) or Claude, or Codex ……

- Enable TMDL extension

- Enable PBIP (TMDL + PBIR) and save

- Link your repo with your PBI workspaces

- Deployment pipelines (PBI) optional

Message: *using native approach is quick and gives much of the benefit!*

Deployment Pipeline with Native PBI GIT Sync

PBI Workspaces
WITH DEPLOYMENT PIPELINE

TEST workspace → UAT workspace → PRD workspace

Native sync with GIT

CD

Azure DevOps
GIT repo

Create feature branch from main

Main branch

CI Publish to main + Pull Request

Feature Branch (in VS Code)

PBI Projects File (PBIP)
TMDL + PBIR

Source control

BPA

Code review

Automation

Agentic AI

Metadata

Basic Setup

Workspace settings

General

License info

Azure connections

System storage

Git integration

OneLake

Workspace identity

Outbound networking

Inbound networking

Encryption

Monitoring

Git integration

Connect to Git to manage your code and back up your work. Learn more

ℹ️ **Preview items** Some item types are only available in preview when using Git. Learn more ⧉

Connect Git provider and account

Git provider

| Azure DevOps | GitHub |

Connect Git repository and branch

Comparing Copilot plans 🔗

The tables below show the features available in each Copilot plan.

	Copilot Free	Copilot Pro	Copilot Pro+	Copilot Business	Copilot Enterprise
Pricing	Not applicable	$10 USD per month, or $100 USD per year (free for some users)	$39 USD per month, or $390 USD per year	$19 USD per granted seat per month	$39 USD per granted seat per month
Premium requests	50 per month	300 per month	1500 per month	300 per user per month	1000 per user per month

---

Changes 4 •   Updates

Commit message

Add details before commiting or we'll add for you by default.

☑ 📄	Item	Status
☑	Analysis	⚡
☑	Analytics v2	⚡
☑	Analysis	⚡
☑	Analytics v2	⚡

⬦ Commit   ↩ Undo

🔷 Test 🔷

+ New   ⬆ Upload   🔗 Create deployment pipeline   ⊞ Create app   👥 Manage access   ⋯   ◈ Source control   🔍 Fi

📄	Name				Git status	Type	Owner	Refreshed	Next refresh	Endorsement
📊	All visual types v3	🔗	☆	⋯	✅ Synced	Report	RebublishTest	03/01/23, 9:44:25	—	—
⊞	All visual types v3				✅ Synced	Dataset	RebublishTest	03/01/23, 9:44:25	N/A	—

Our Journey

- >40 models, >90 reports, 7 people
- Not native approach – automate pipeline approach for CI / CD through Azure DevOps
- 1-month planning, 2-month development, 1-month migration + training
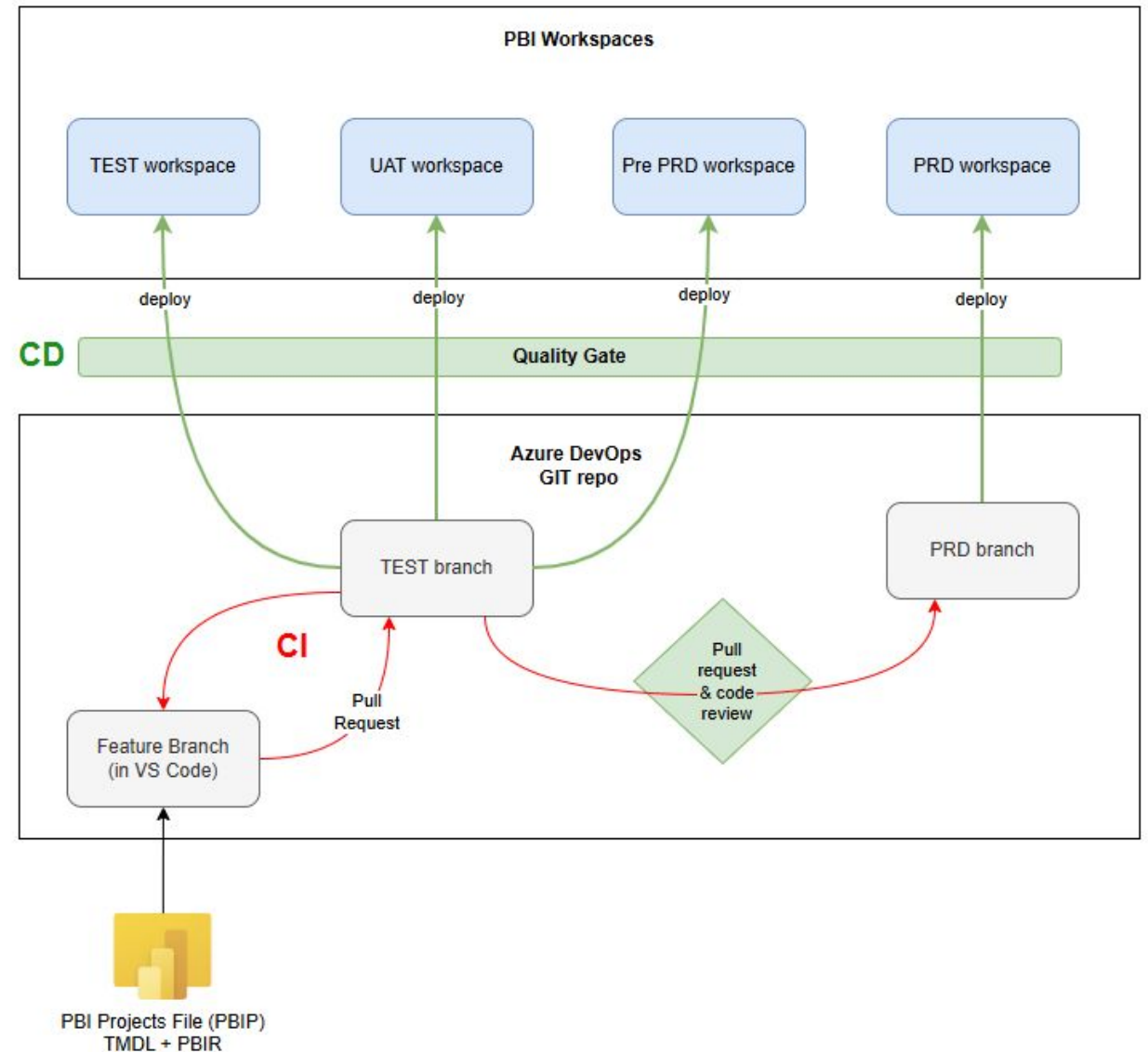- Takes time to learn, things can be a little new and scary, but at the end you are at a much better place.

Our Setup

- Two branches + feature branches
- VS Code + Azure DevOps (git)
- Help from Senior Engineer!
- CI pipelines to merge
- CD pipelines to deploy
- Quality Gates (BPA) on deploy
- Documentation + Metadata
- Copilot for GitHub (agentic AI)
- Helper scripts (python / PS)

FUTURE:
Model Context Protocol (MCP) Servers
Custom Graph DB + BPA, metadata & data
edits

Integrating AI into your workflow

Level 1 – Assistive Use (Chat-Based)
AI tools like ChatGPT used in a browser

Level 2 – Agentic Integration
AI runs inside your environment (e.g., VS Code with Copilot or local agent). Can edit scripts, create files, and trigger task, includes some CLI.
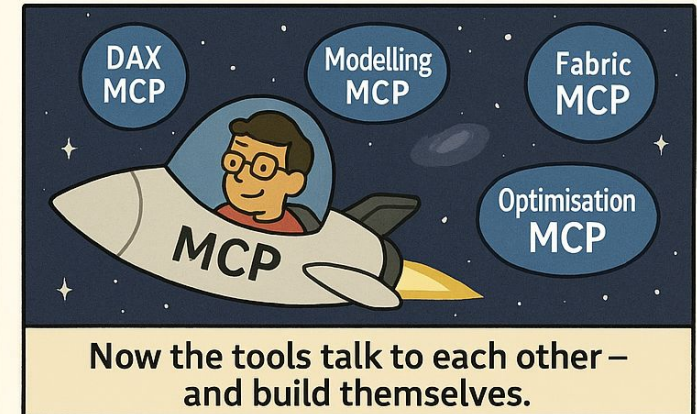
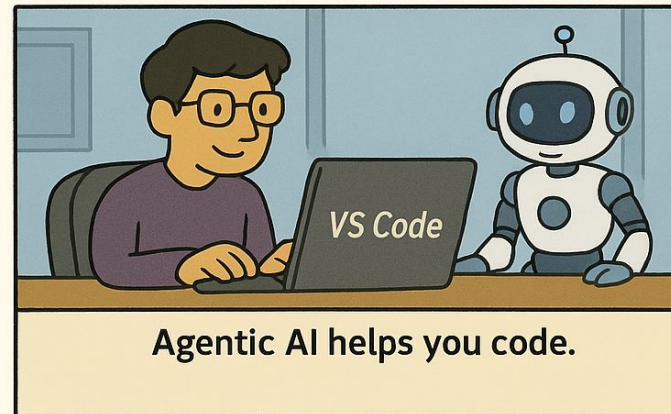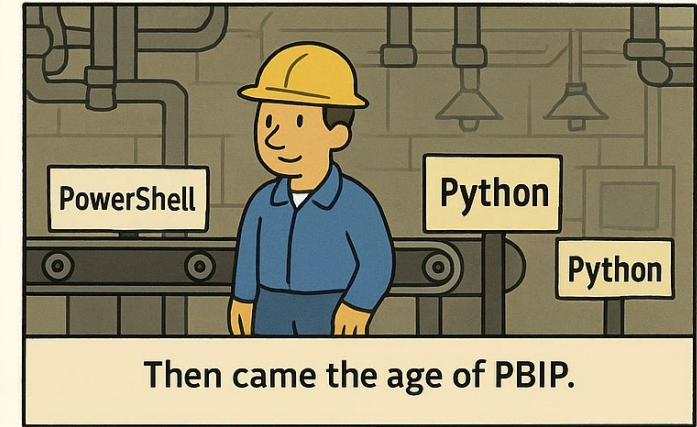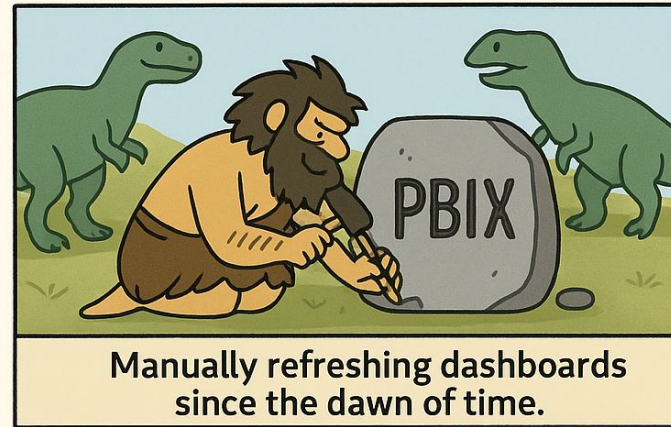**Level 3 – Tool Augmentation (MCP / Plugin Layer)**
AI gains structured access to your tools through the **Model Context Protocol**
-   MS Official MCP's in development

Level 4 – Systemic Integration (CLI + Pipelines)
The CLI becomes the automation backbone connecting components. Pipelines and scripts can call the CLI, and AI can invoke it to perform real actions.

Level 5….. Who knows?

Manually refreshing dashboards since the dawn of time.

Then came the age of PBIP.

Agentic AI helps you code.

Now the tools talk to each other – and build themselves.

Demo

- Create feature branch
- Show repo structure
  - TMDL + PBIR structure
  - Configs
- Automation
  - BPA (show BPA report)
  - Auto Fixer
- GitHub Copilot
  - Descriptions prompt
  - Visualising relationships
- PR to branch + Deploy