DataKitchen

The FITT Way To Data Products:

A New Data Architecture For A Product-Centric World

October 9, 2025

Before We Start

Christopher Bergh
CEO, Head Chef
DataKitchen

Today's recording and slides will be shared after the meeting.

Put your questions in the chat window; we will answer questions at the end of the session (or during!).

We have targeted 45 minutes for our presentation.

Agenda

Data Products

Background and Motivation

Perspective On Data Engineering

What Is FITT (Functional, Idempotent, Tested, Two-stage) Data Architecture?

How to get FITT.

Current Discussion of Data Products …

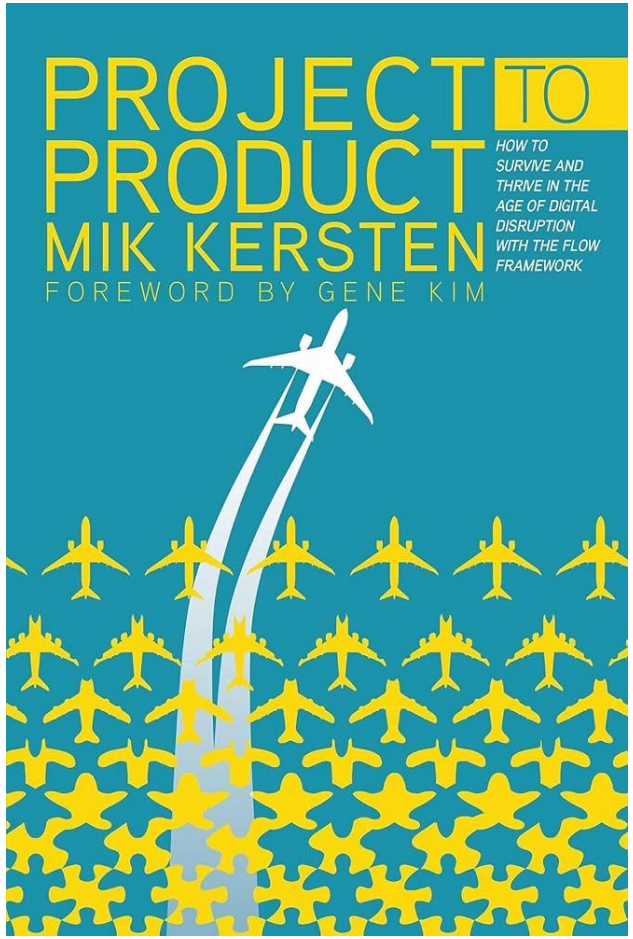
(From ChatGPT)

A data product is a reusable, trusted, and well-defined data asset—such as a dataset, metric layer, feature set, or dashboard

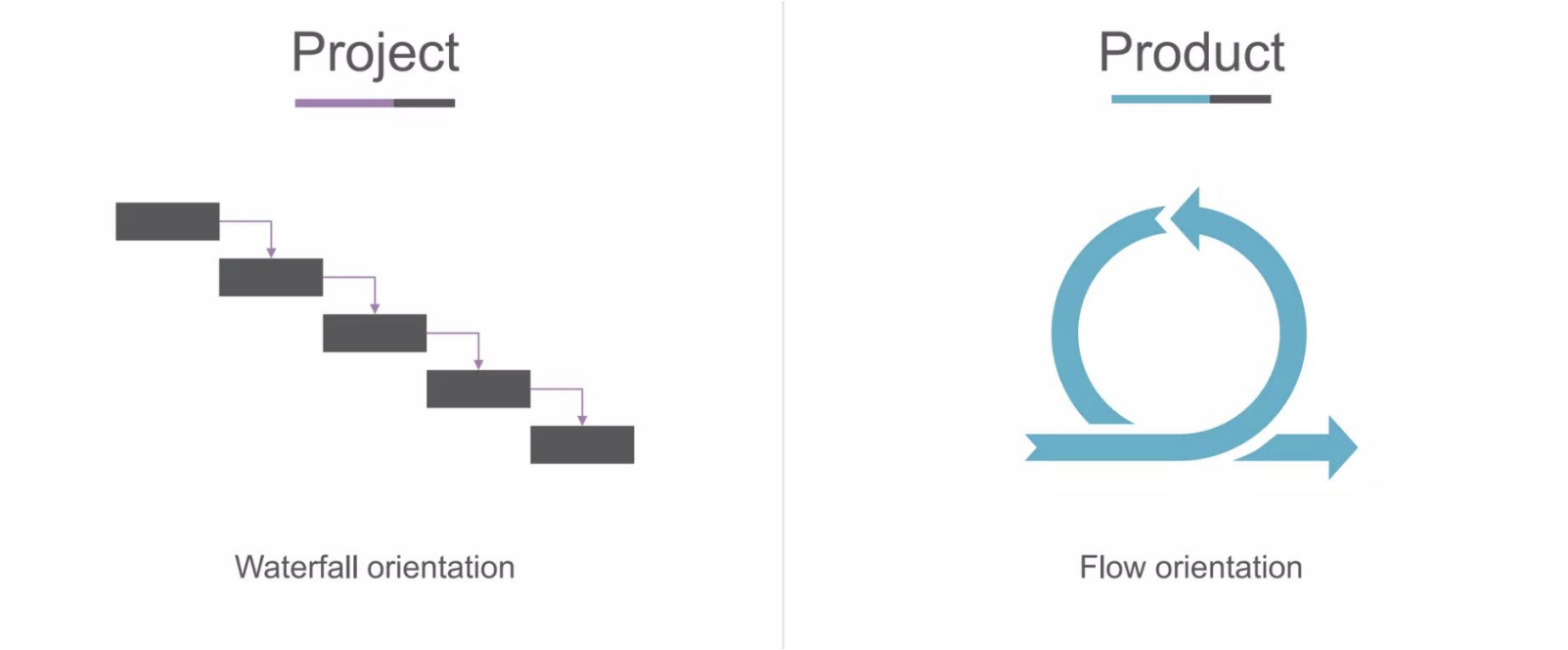
…. that is designed, built, and managed like a product.

Data Products Are A ‘How’ Not A ‘What’

- **Data Products are a method, not a component list**
  - They do not have an end (like a project)
  - They are iteratively improving
  - Are focussed on delivering value, not meeting budget
  - Outcome oriented, not data oriented
  - Can be any type of deliverable
  - Success is defined by customer satisfaction, not on time delivery or internal process followed.

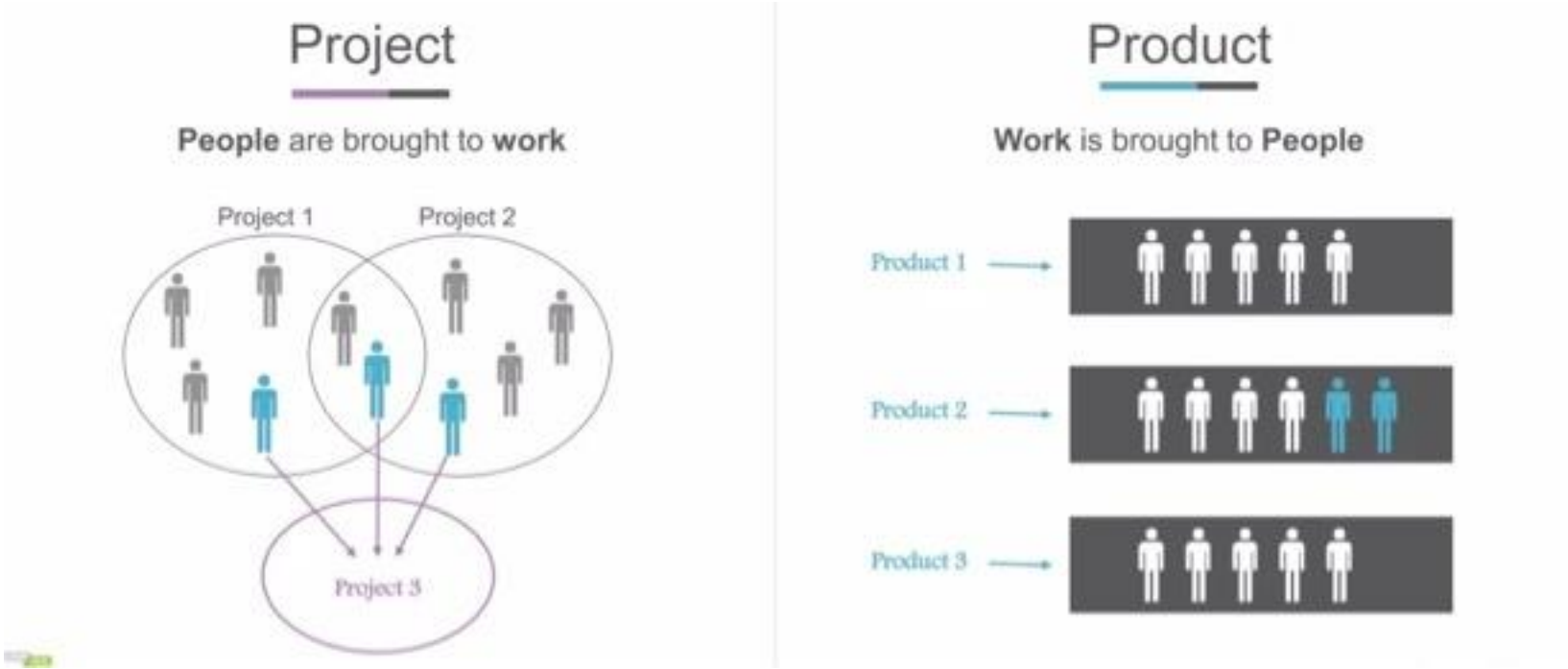


Data Projects to Data Products



From Tasktop

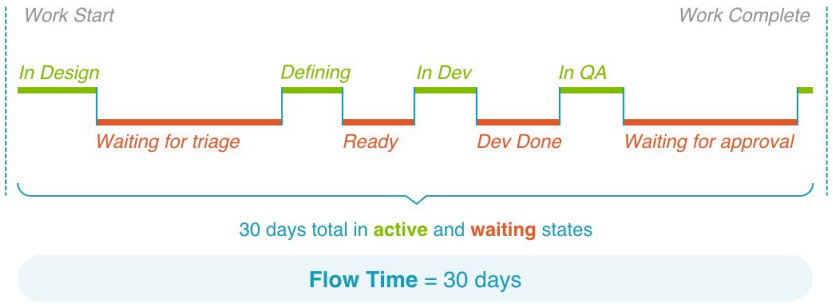
Data Projects to Data Products



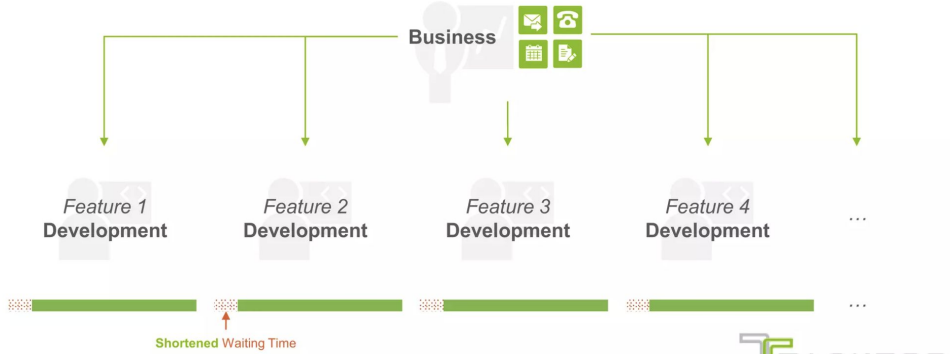
From Tasktop

Data Projects to Data Products

Project



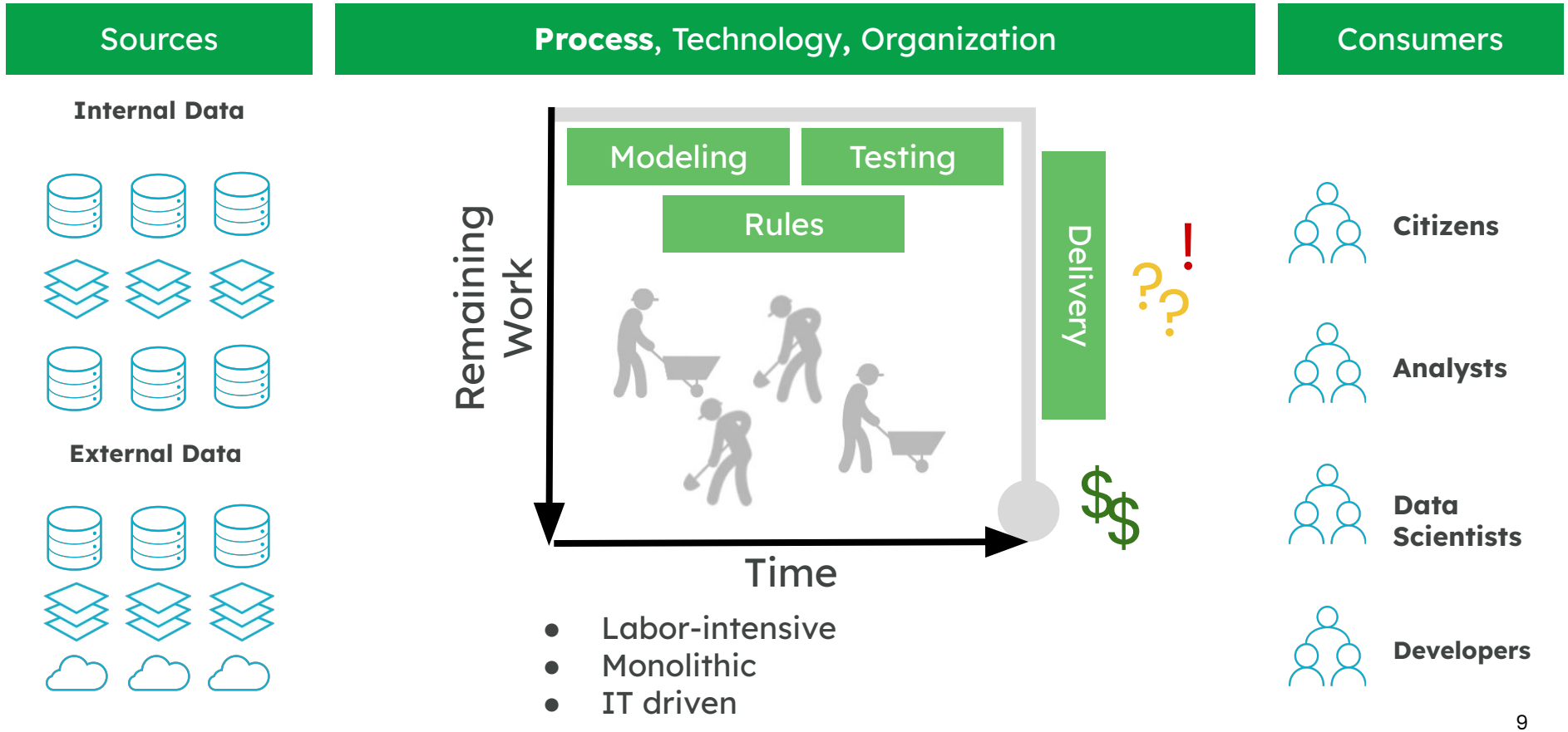
Product



From Tasktop

DataKitchen

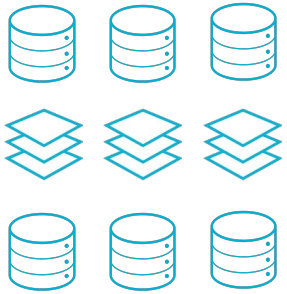
Data Products – The Wrong Way



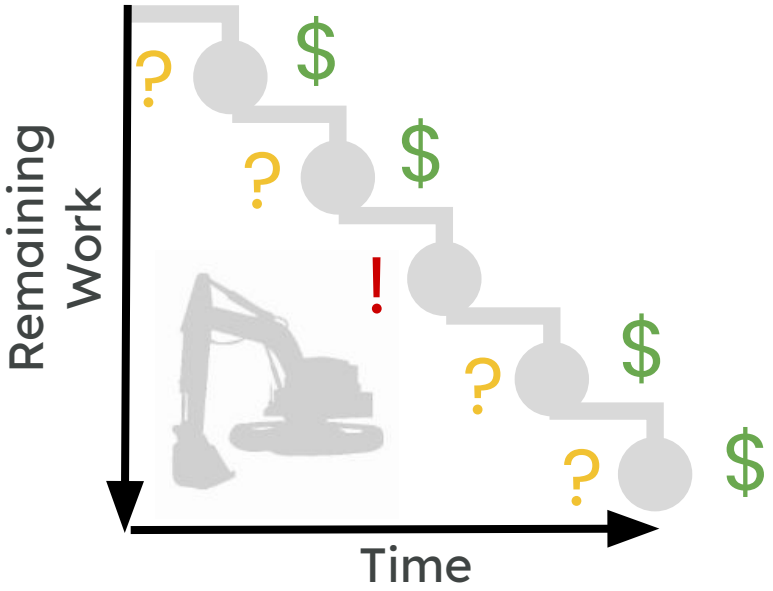
Data Products Are Continuously Improved

Sources	**Process**, Technology, Organization	Consumers

Internal Data



External Data



- Automated
- Incremental
- Collaborative

Citizens

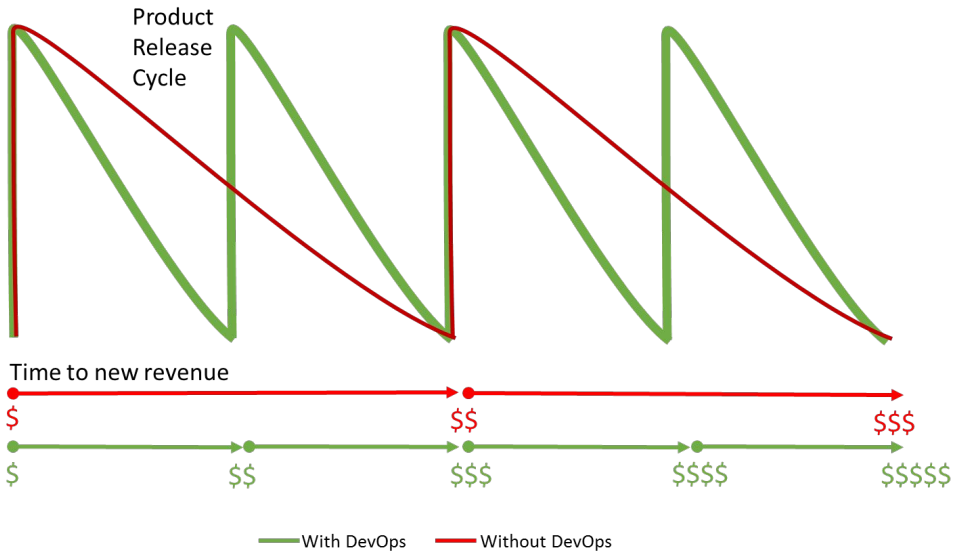
Analysts

Data Scientists

Developers

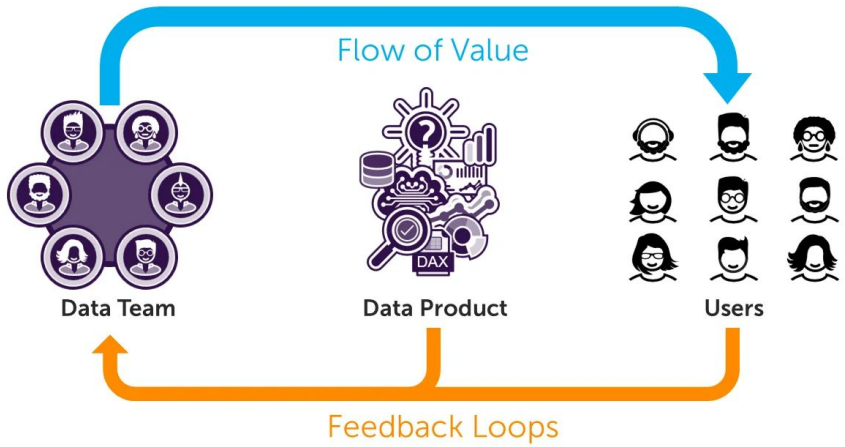
DataKitchen

Data Products Are Never Done



- Faster releases of analytics products
- Increased learning of your team
- Better insight
- Faster and more value for your data analytics

Data Products Maximize Team **Flow**



Improving and measuring the 'flow' of value to your customer is the #1 goal of Data Products

- What is the 'flow' of value to your customer?
  - New data sets
  - Improved models/visualizations/data
  - New code/configuration
- Key measurements
  - Cycle time of deployment ('flow time)
  - Success Rates of Deployment ('flow efficiency')
  - Work done ('flow velocity')

Agenda

Data Products

Background and Motivation

Perspective On Data Engineering

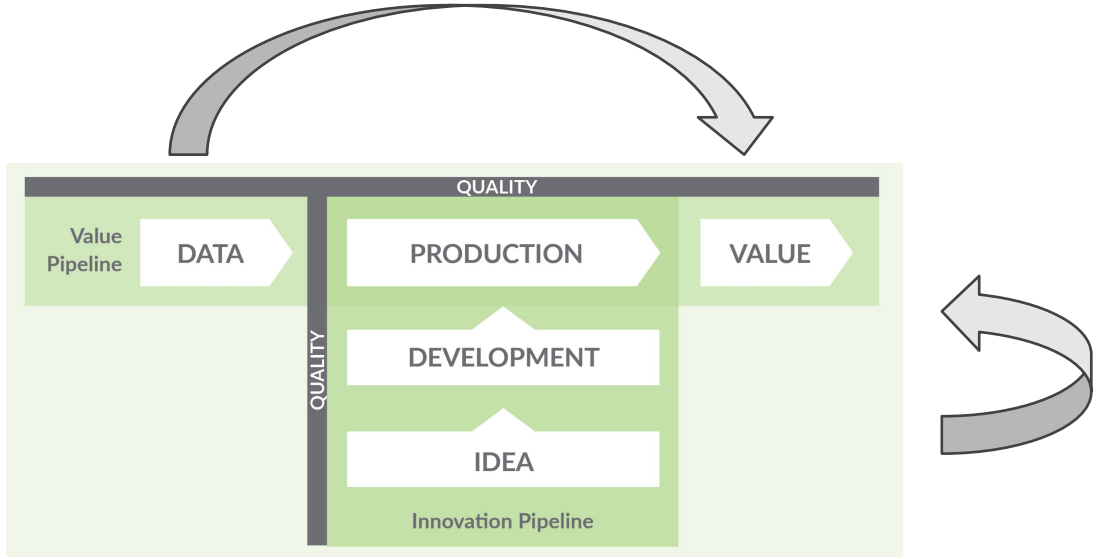
What Is FITT (Functional, Idempotent, Tested, Two-stage) Data Architecture?

How to get FITT.

Challenges of Data Products

- **There are three main challenges with applying a Data Product mindset to data teams:**
  a. You get crappy data and sh** breaks.
  b. Your customers don't know what they want until they see it.
  c. You always have too much to do.
- **For the last dozen years we (DataKitchen) are trying to fix this problem. 'DataOps'**

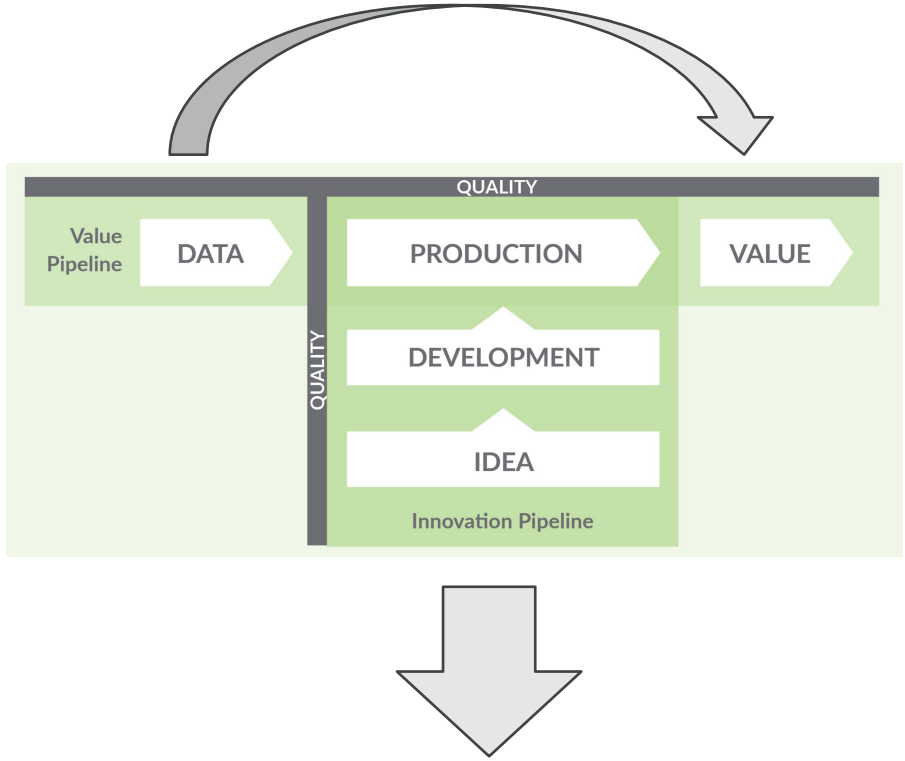
DataOps



DataKitche

DataOps

Value Pipeline
Decrease Production Errors



Innovation Pipeline
Increase Development Cycle Time

Result: Less Waste, 10x Team Productivity Improvement, Improved Trust

DataKitche

Background and Motivation

- There are three main challenges with applying data product mindset to data teams:
  - You get crappy data and sh** breaks.
  - Your customers don't know what they want until they see it.
  - You always have too much to do.
- For the last dozen years we (DataKitchen) are trying to fix this problem. 'DataOps'

- We have build a profitable, independent business while doing data engineering consulting with our enterprise software
  - **We delivered Data Products following DataOps principles and FITT architecture with all our consulting customers.**
  - **Result: 10x productive, miniscule low errors, fast new development cycle time**

Agenda

Data Products

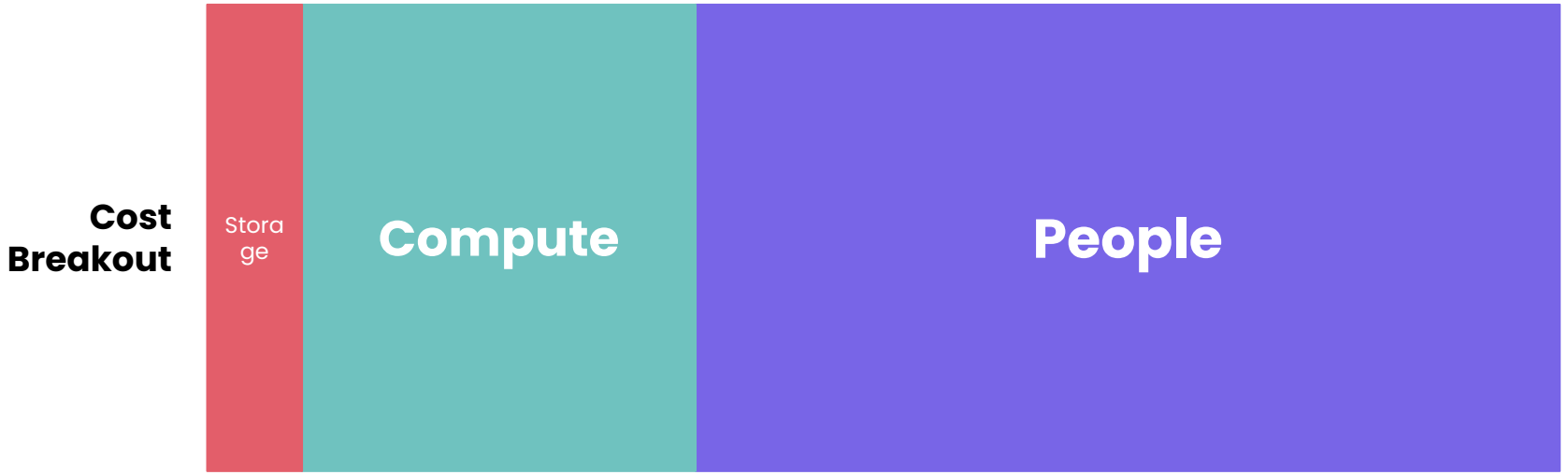
Background and Motivation

Perspective On Data Engineering

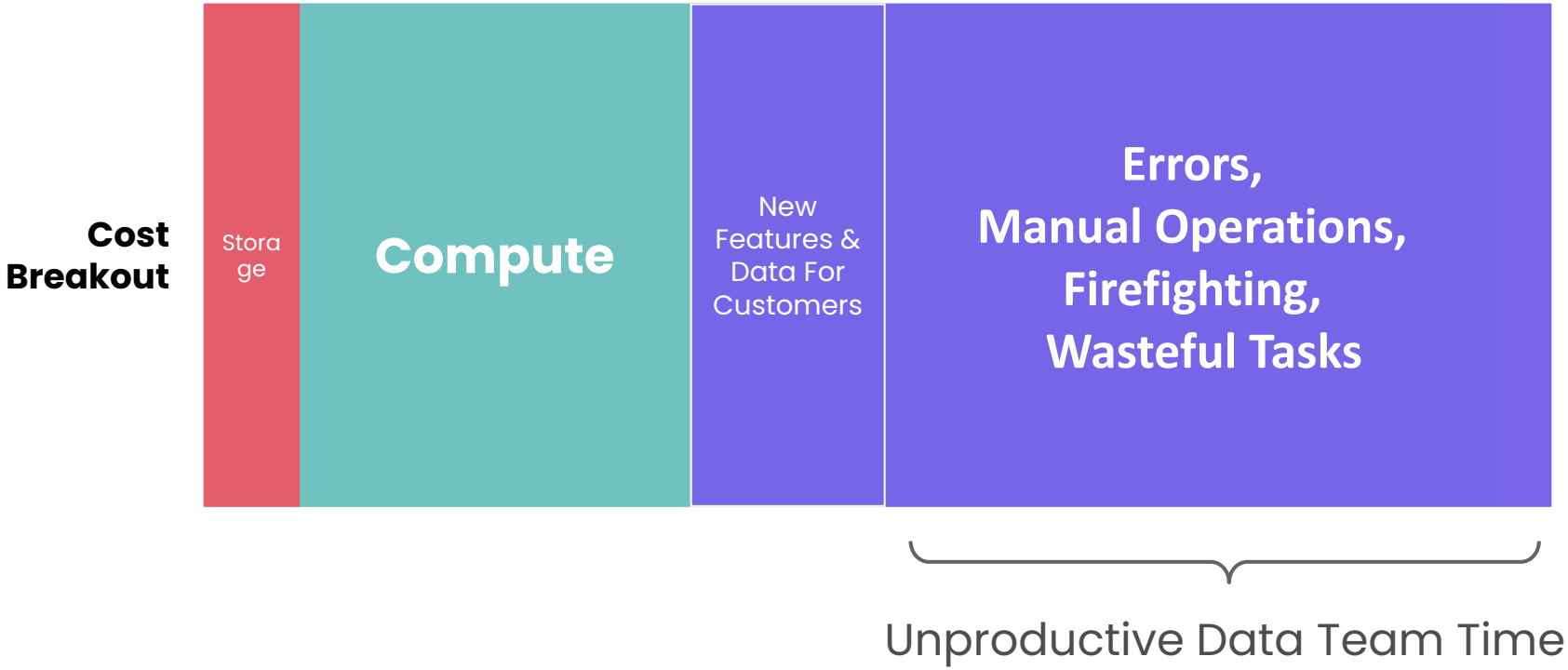
What Is FITT (Functional, Idempotent, Tested, Two-stage) Data Architecture?

How to get FITT.

Data Engineering time (= cost) >> compute costs > storage costs.

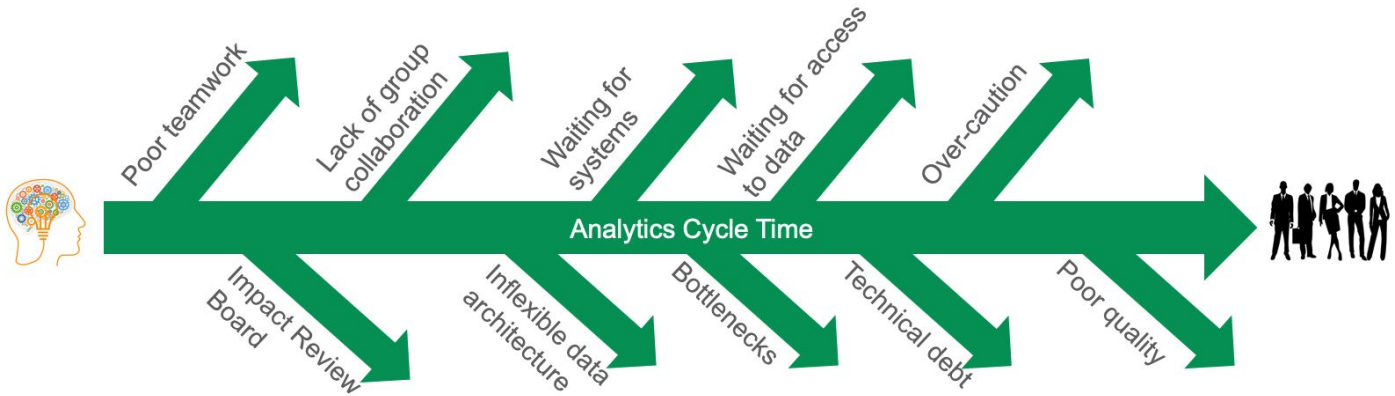


Unfortunately,
The Majority Of Data Team Time Is Wasted

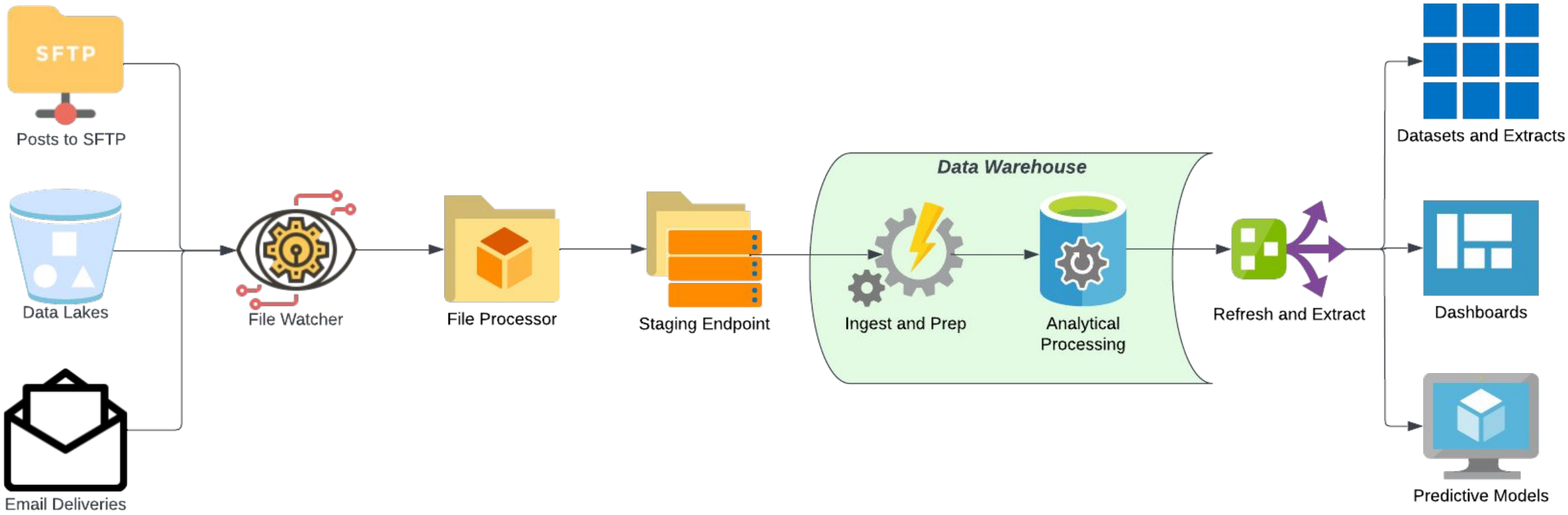


Maximize Flow:

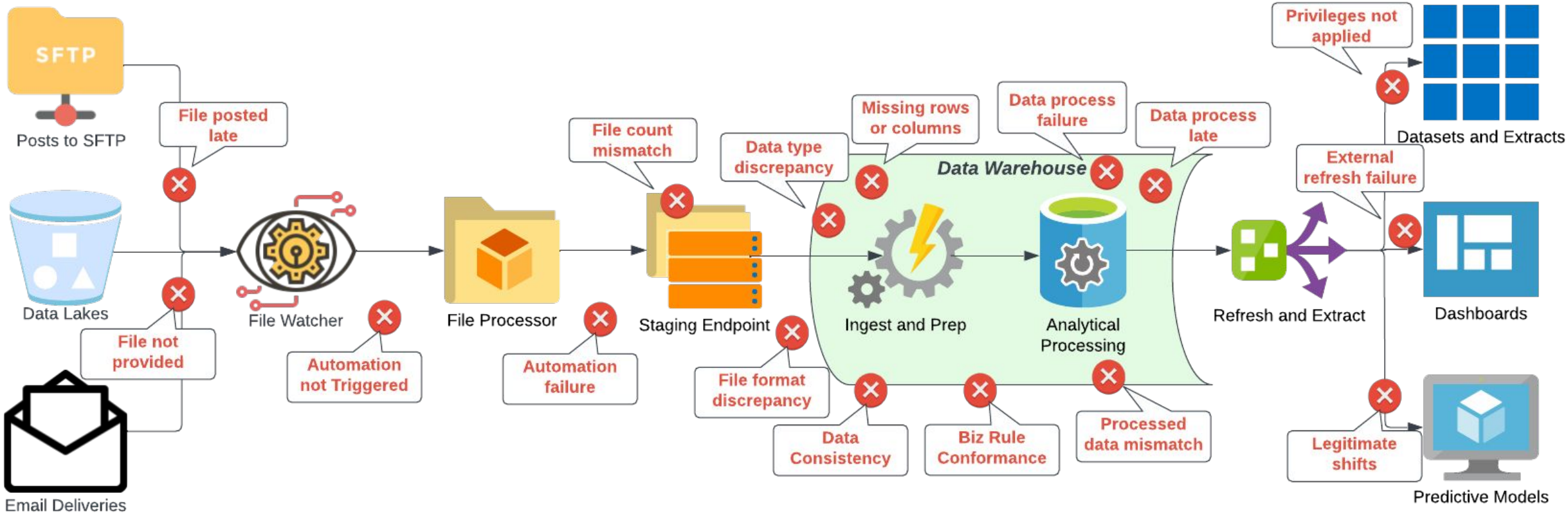
Data Architecture Should Focus On Maximizing Data Engineer Productivity



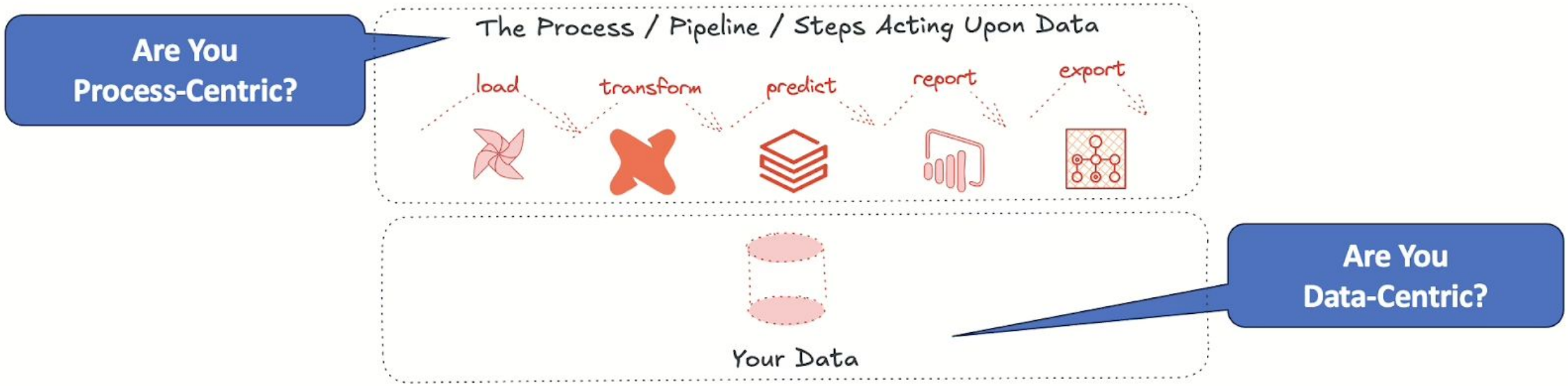
Data Processes …



Data Processes Have Lots Of Potential Errors



Process Acting On Data >>> Data Itself.



https://datakitchen.io/unlocking-data-team-success-are-you-process-centric-or-data-centric/

Process Acting On Data >>> Data Itself.

"We realized that the true problem, the true difficulty, and where the greatest potential is – is building **the machine that makes the machine**. In other words, it's building the factory. I'm really thinking of the factory like a product." – Elon Musk

"If any engineer has to choose between working on a feature or working on dev productivity, **always choose developer productivity**." – Satya Nadella

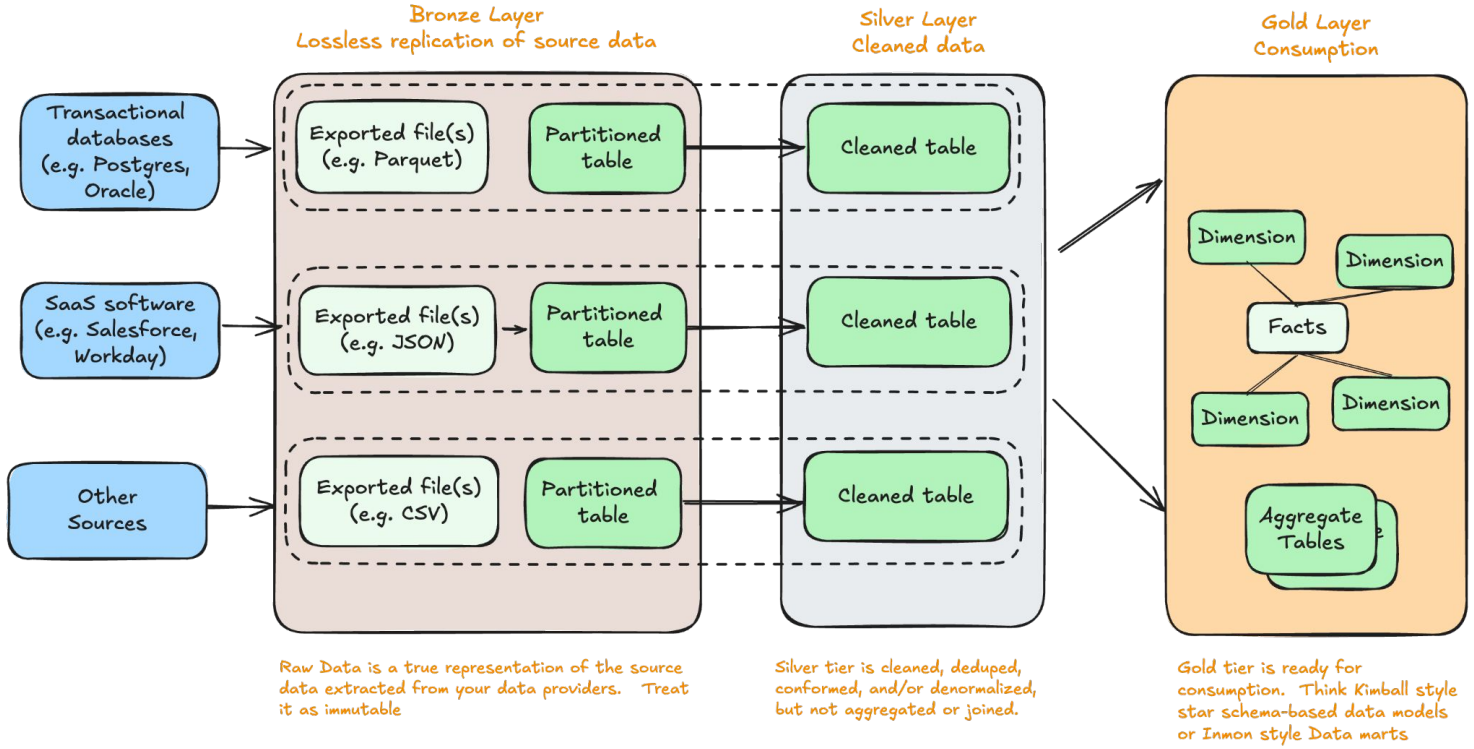
Google

"At Google, we have over **2,000 engineers who contribute to Engineering Productivity**."

https://datakitchen.io/unlocking-data-team-success-are-you-process-centric-or-data-centric/

Medallion Data Architectures Are Designed To Maximize Data Vendor Revenues …

Not Data Team Productivity.



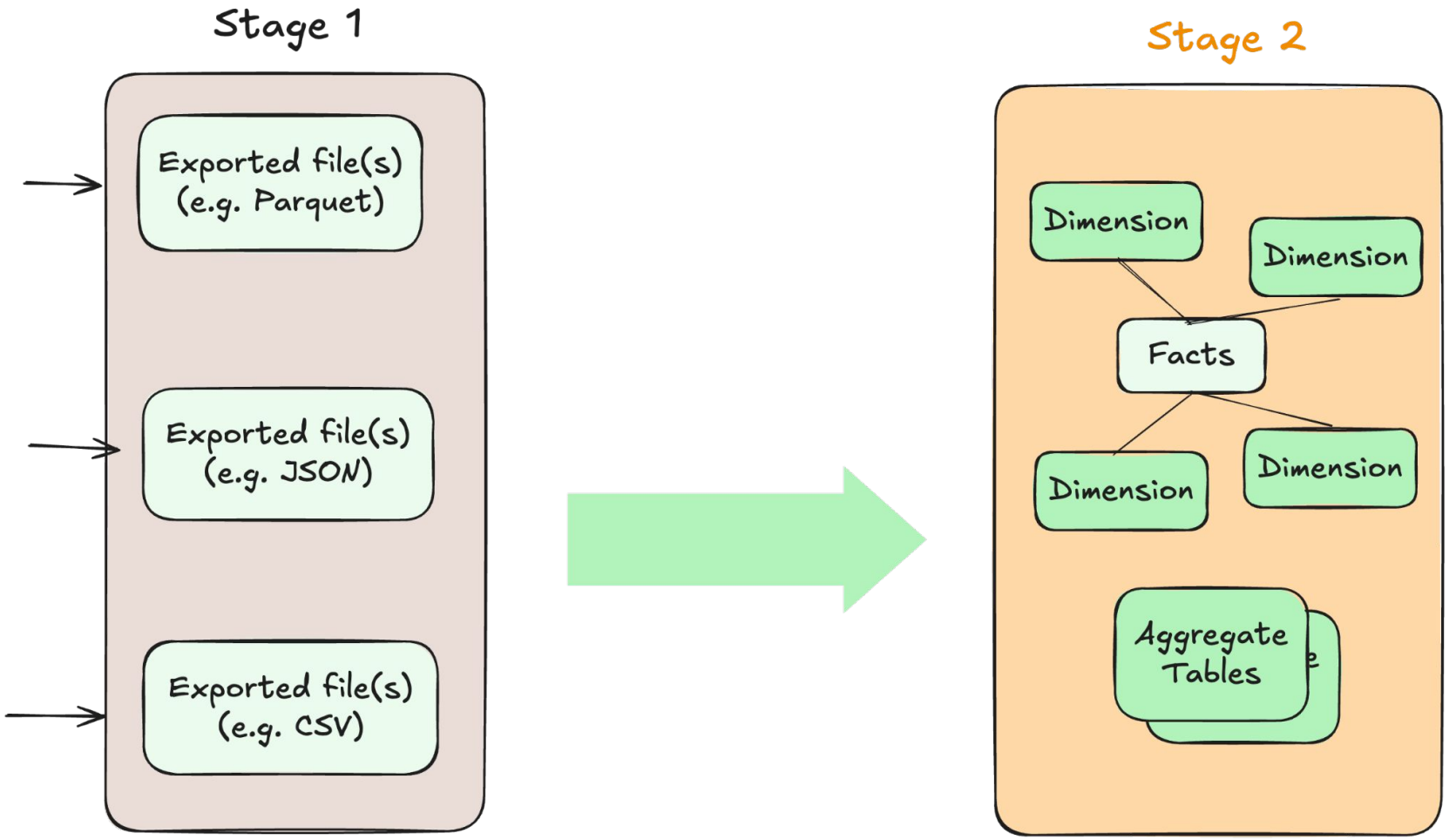
An example of a medallion architecture with three zones: Bronze, Silver, and Gold

Agenda

Data Products
Background and Motivation
Perspective On Data Engineering
What Is FITT (Functional, Idempotent, Tested, Two-stage) Data Architecture?
How to get FITT.

What's A Better Data Architecture?

FITT:
Functional,
Idempotent,
Tested,
Two-stage



FITT Is Not A New Idea, Just **Sound Engineering Principles** Applied to Data and Analytic Systems

FITT Is Designed Maximize **Flow** and Minimize **Errors**

FITT Really Shines When

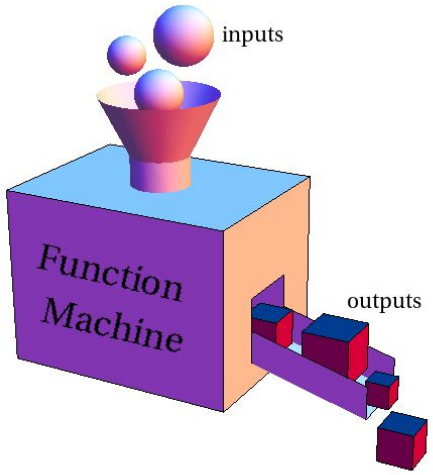
- Your total data size data isn't massive (under 100TB-ish, with much smaller updates)
- Data updates/refreshes happen hourly or longer
- You're doing serious data integration work (building star schemas, complex joins, predictive models, the whole nine yards)
- Business stakeholders keep asking for pipeline changes (because of course they do)
- You need results you can actually trust (and honestly, who doesn't?)

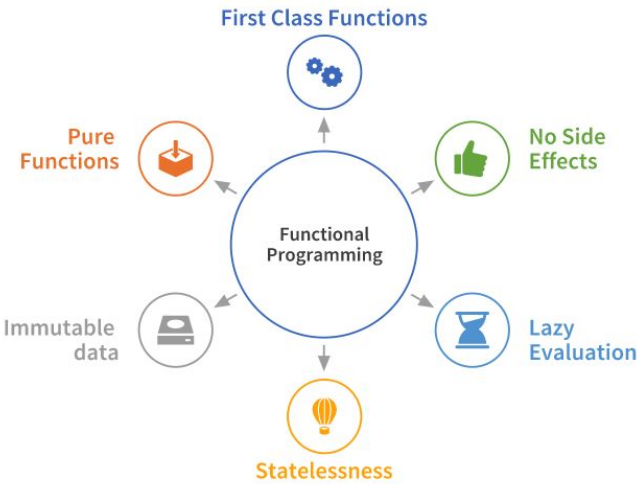
Functional

What? For every version of code, each transformation 'chunk' takes input and produces output, period.

No more mystery dependencies or hidden state lurking in your transformations.

For every version of code, each transformation takes clearly defined inputs and produces predictable outputs, period.





Idempotent

What? It is running a data pipeline multiple times with the same input will always produce the same output.

- When everything is idempotent, recovery becomes trivial.
  - Pipeline broke? Just re-run it.
  - Weird results? Re-run it.
  - Want to test a change? Re-run it on yesterday's data.
  - Need to backfill from 80% done. Just run the substeps.
- The psychological impact is enormous.
- "build a little, test a little, learn a lot" development rhythm
  - Iterate on individual components without having to rebuild entire pipelines.

Tested

“Our 10,000 tests didn’t catch it. Are you sure you found a problem?”

What? Treat tests like first-class citizens. All automated tests.

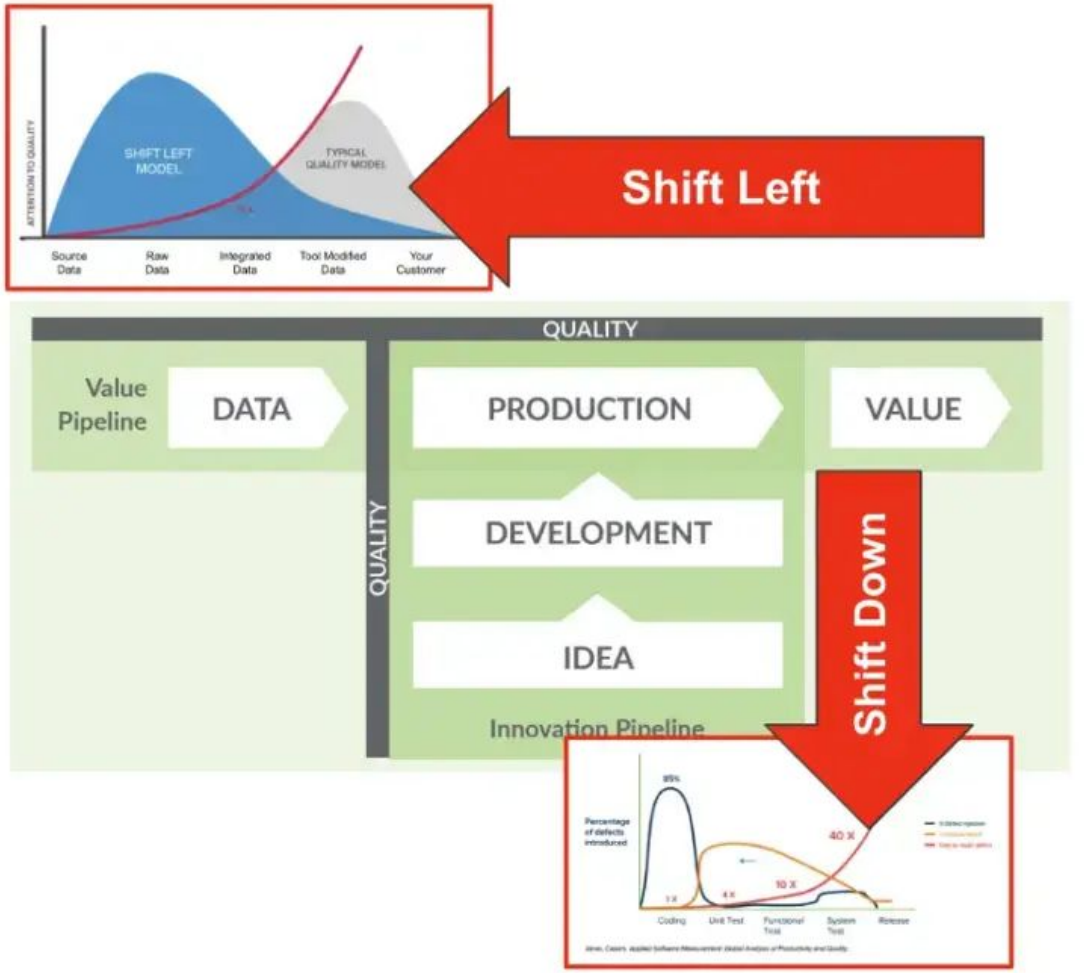
Tests are living proof of what the system should do. And by ‘tests,’ we mean numerous tests —hundreds, thousands, covering every table.

Test the data and test/monitor tools acting upon data

Tests are the gift you give to your future self.

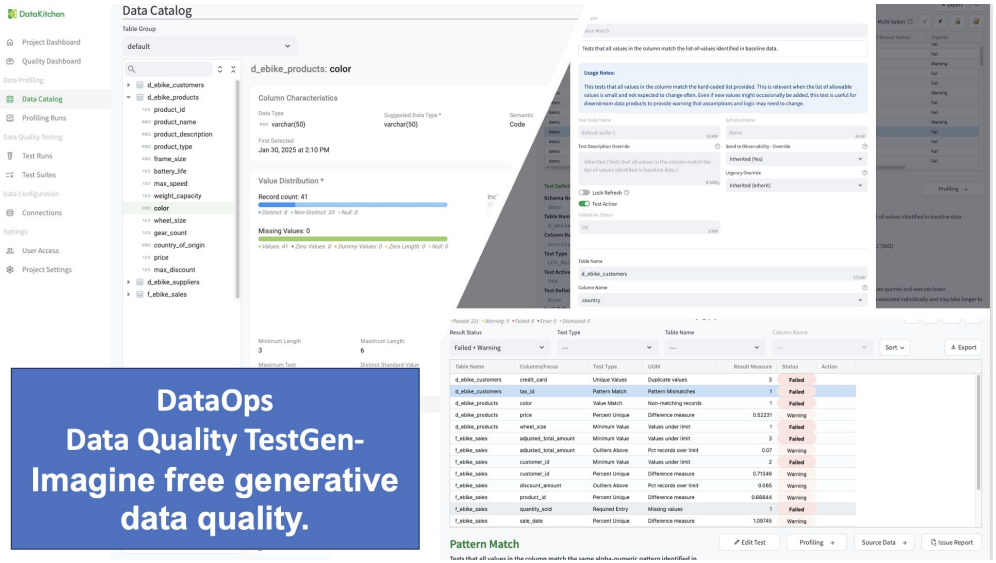
Build ‘Andon cords’ into production workflow – stop on errors

Test Coverage Is Hugely Important In FITT Data Architectures



https://datakitchen.io/scaling-data-reliability-the-definitive-guide-to-test-coverage/

[AD] DataKitchen TestGen Software:
80% of the data tests you need automatically



Open Source DataOps Data Quality TestGen:

- Full Featured, Data Quality Tool
- In Database Execution
- Full Featured (UI, AI, Rules) One User
- Enterprise Version starts at $100 per user per month

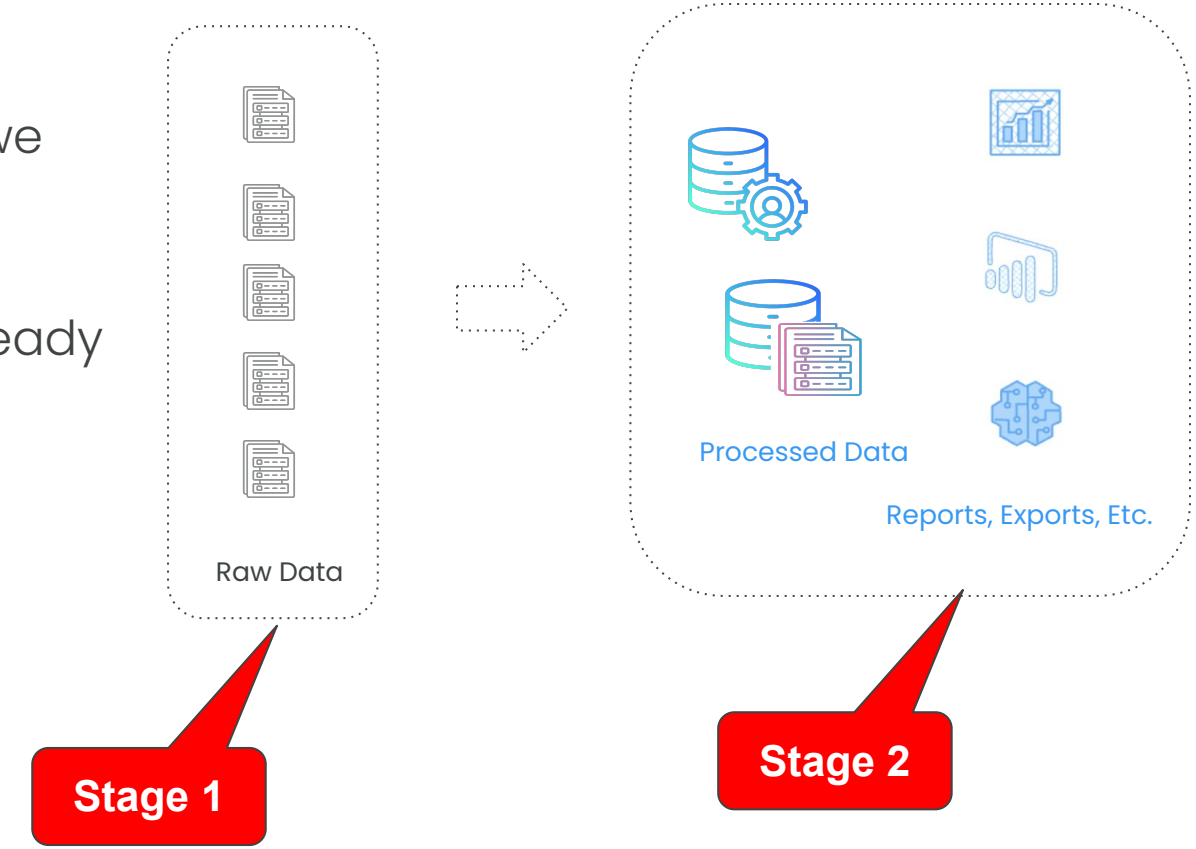
It Does Five Tasks:

1. Data Profiling
2. Dataset Screening And Hygiene Review
3. AI Generation of Data Quality Validation Tests, Custom Tests
4. Data Testing and Anomaly Detection
5. Data Quality Scoring Dashboards

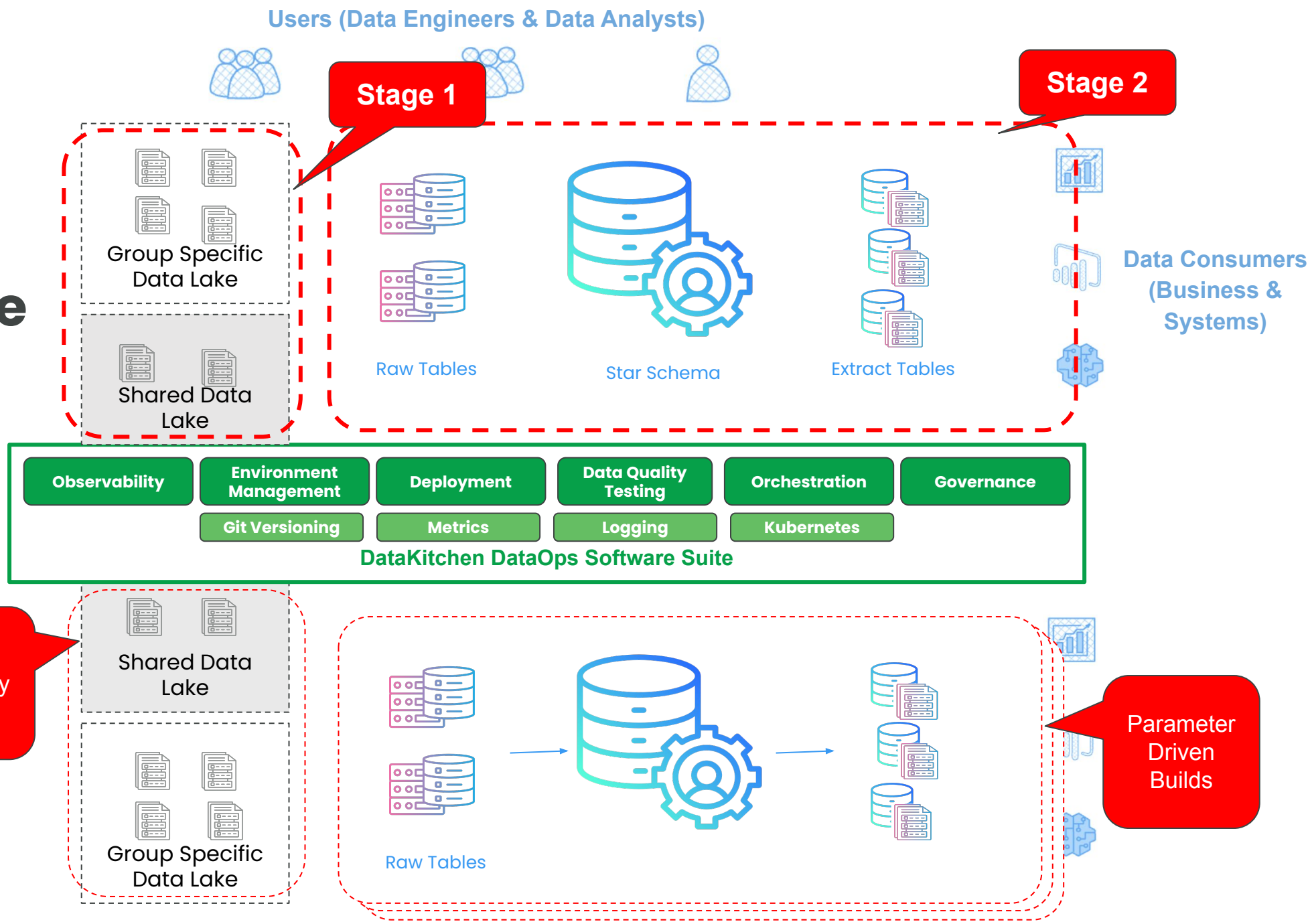
Two Stage

Production

- Raw Data, which represents exactly what we received from source systems, immutable forever.
- Final Data (models, exports, reports, etc), ready for consumption.
- Done in one step (with of course, many substep)
- Everything in between? Can be deleted



Fortune 100 Customer Two Stage Data Architecture



DataKitchen

Agenda

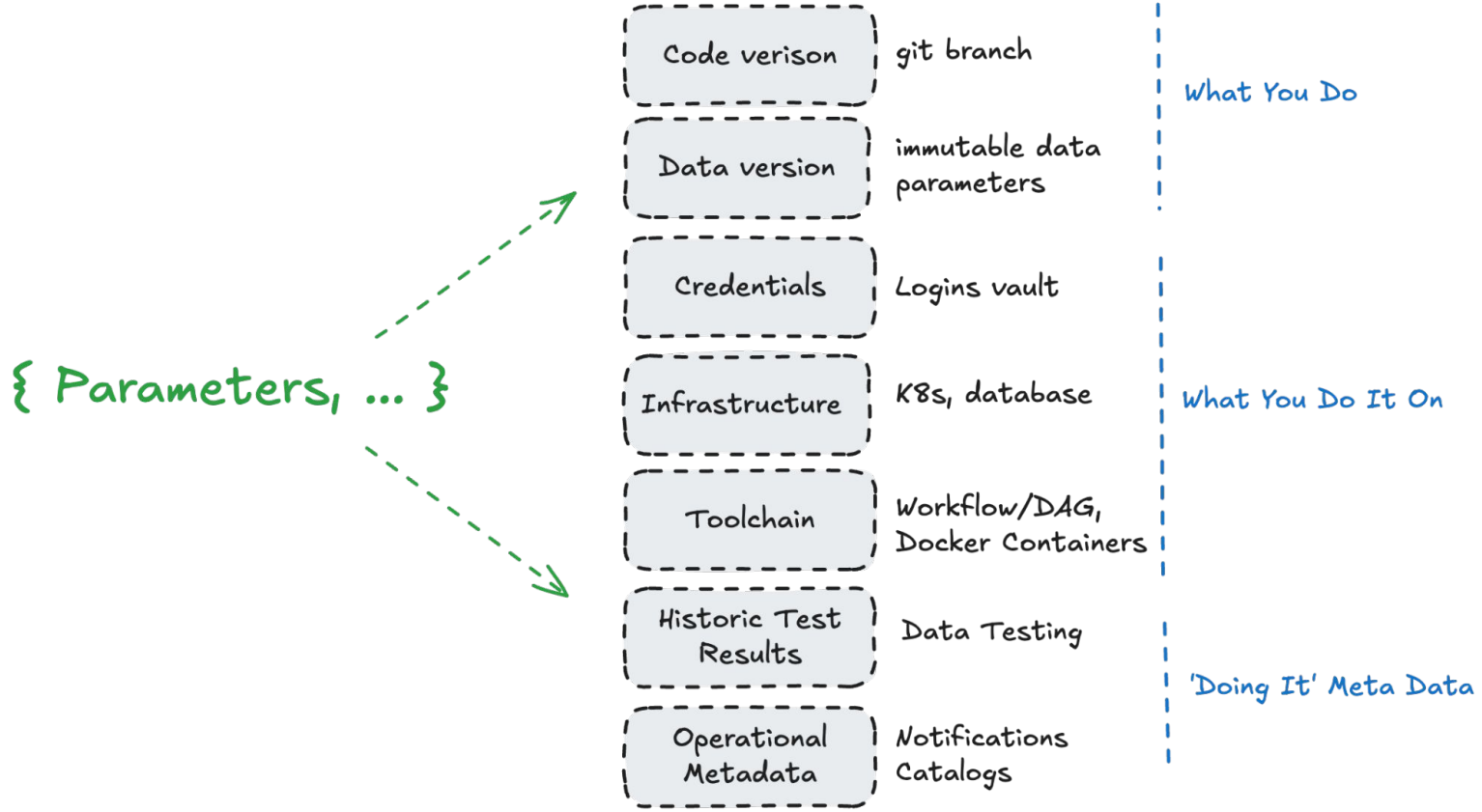
Data Products
Background and Motivation
Perspective On Data Engineering
What Is FITT (Functional, Idempotent, Tested, Two-stage) Data Architecture?
How to get FITT.

How To Get FITT (Data & Pipelines)

- Parameterization and Templates of Pipelines
  - DataKitchen: Kitchens & Recipes & Ingredients
  - Bauplan: Iceberg & Python DSL
- How to Do, e.g.:
  - Dev: Yesterday's data, today's code (many environments, ephemeral)
  - Prod: Today's data, yesterday's code (single environment, stable).

How To Get FITT (Full Parameterization)

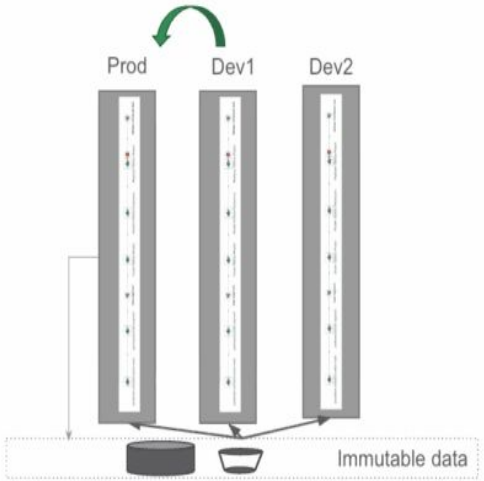
- Parameterize the whole process – be able to run in a new database, on a different git branch, etc. quickly.
- Create a repeatable, reliable process for releasing data analytics (new data set, data science models, schema, code, visualizations …) as one *orchestratable* unit.



How To Get FITT (Environments)

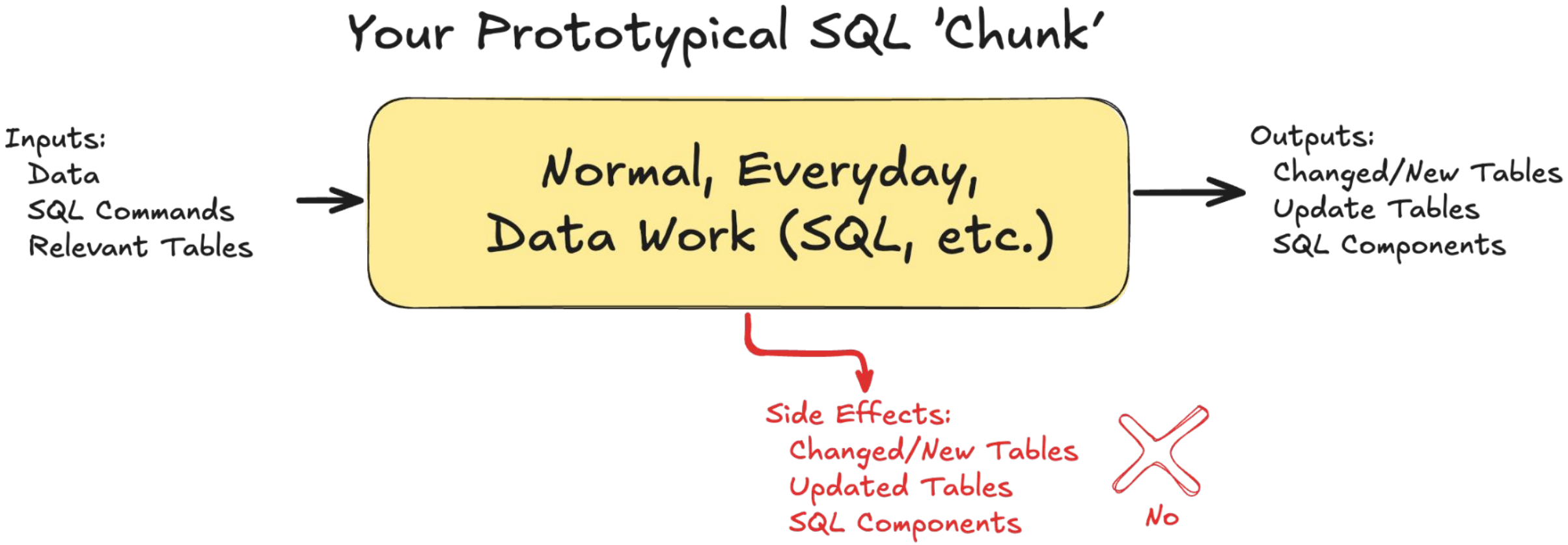
- Script-Driven, Stateless Environment Creation (IaC)
- Environment Parity, Canaries, A/B environments
- Always be able to create a entirely new environment with data—whether that's a new database schema, a separate data warehouse instance, or zero-copy clones in platforms like Snowflake

Each environment has 'state' that needs to be updated/patch as you deploy



Functional Architecture based on Immutable data and pure functions on that data to create results

SQL Not FITT



How To Get FITT (SQL Techniques)

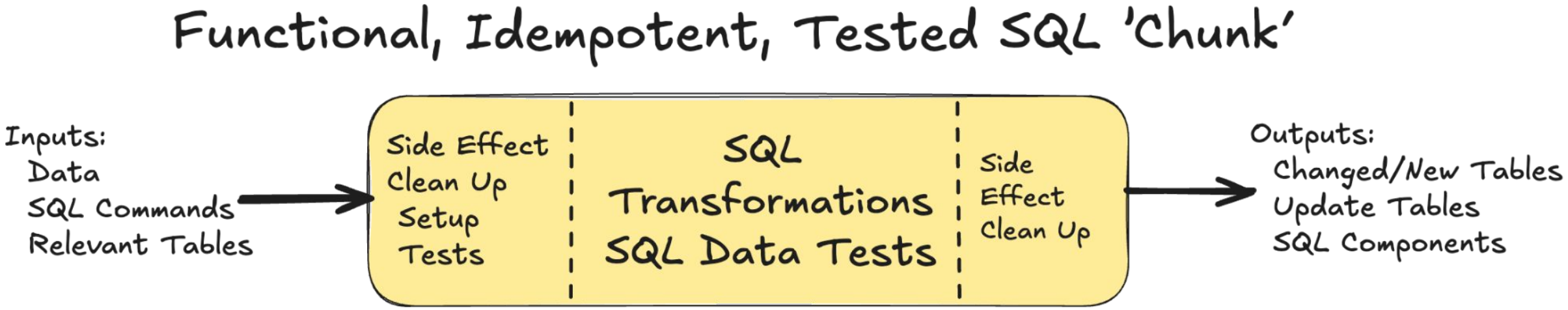
SQL Needs the idea of a Functional Idempotent (FI) Chunk

- A Functional and Idempotent chunk represents a complete unit of data transformation work
- Can be executed repeatedly with identical results regardless of external state.
- Each FI chunk follows specific patterns that ensure functional behavior and idempotent execution within SQL environments: create-new-then-replace pattern
- Tests included

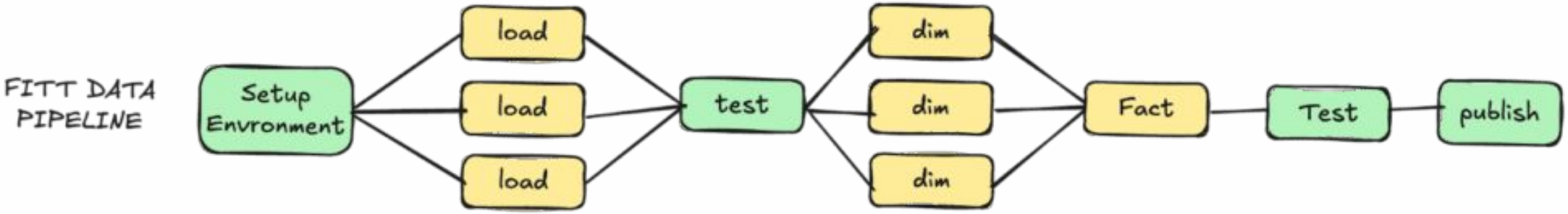
A build process runs multiple FI Chunks

- Always be able to build entire database (and deliverables) from immutable raw data

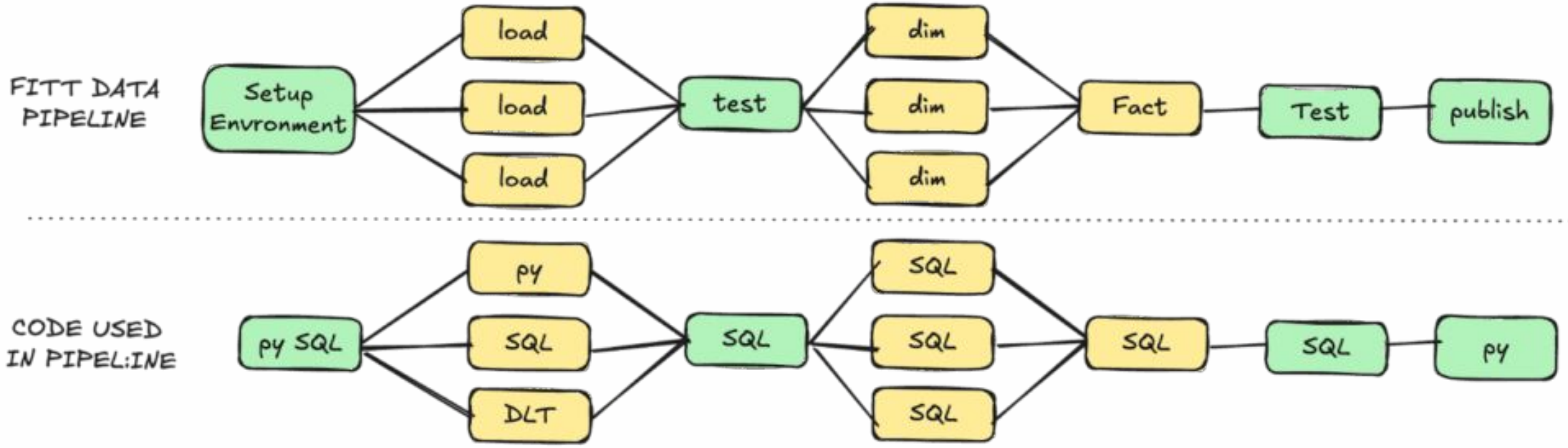
FI Chunk



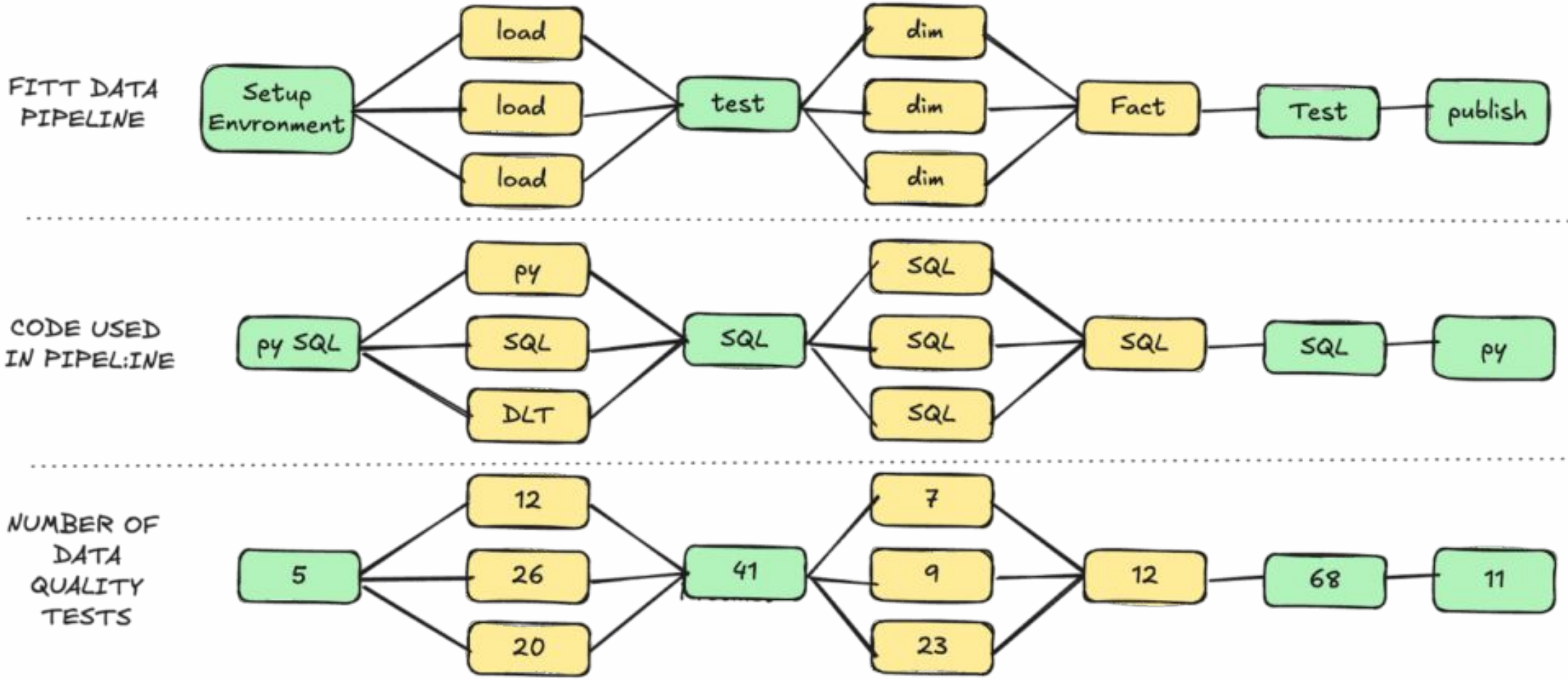
Complete, One Step, FITT Data Pipeline



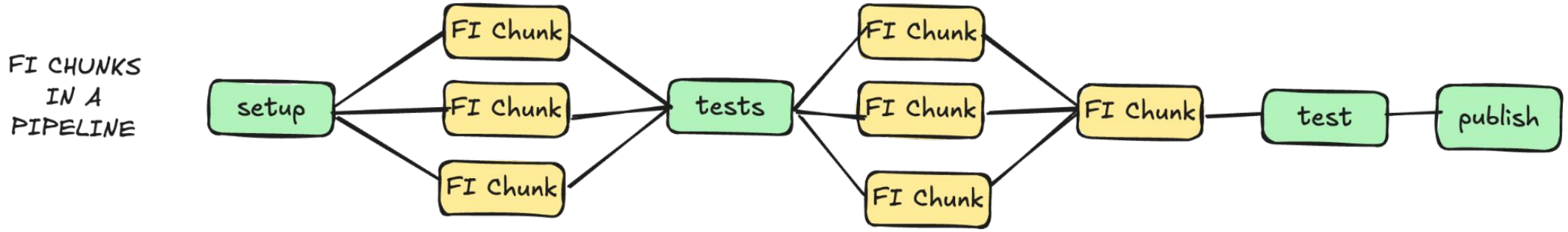
Complete, One Step, FITT Data Pipeline



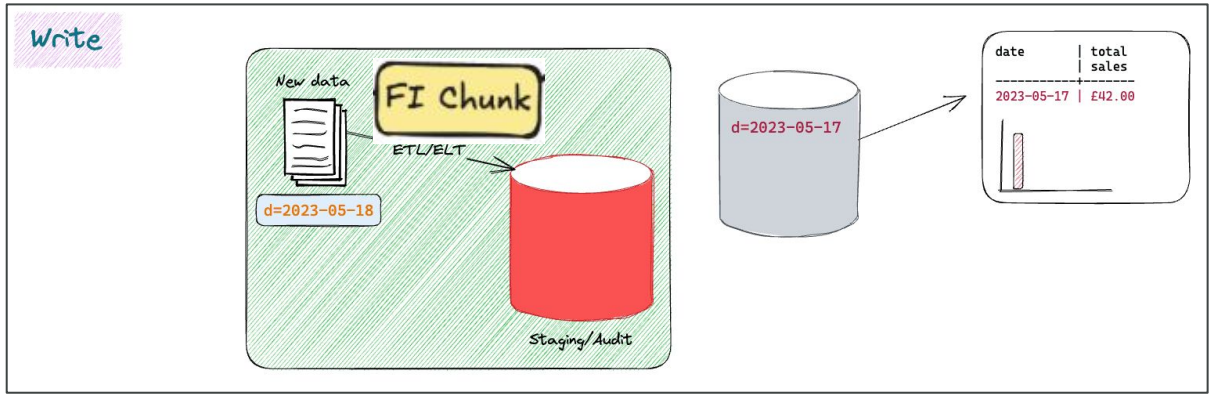
Complete, One Step, FITT Data Pipeline

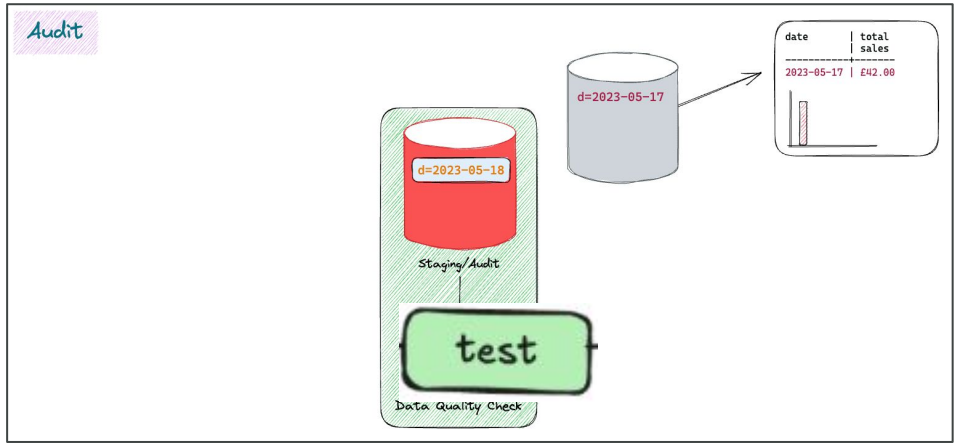


FI Chunk In A Pipeline



Write-Audit-Publish With A 'FI Chunk'

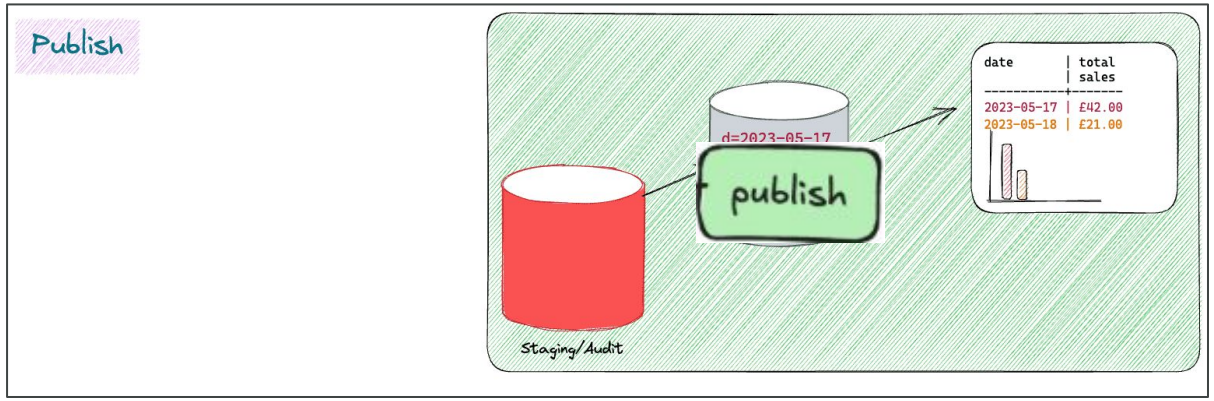




Use widely adopted patterns for keeping bad data out of production systems is [Write-Audit-Publish (WAP).]

WAP is a gatekeeping process that ensures data is tested and approved before it's exposed to downstream consumers.

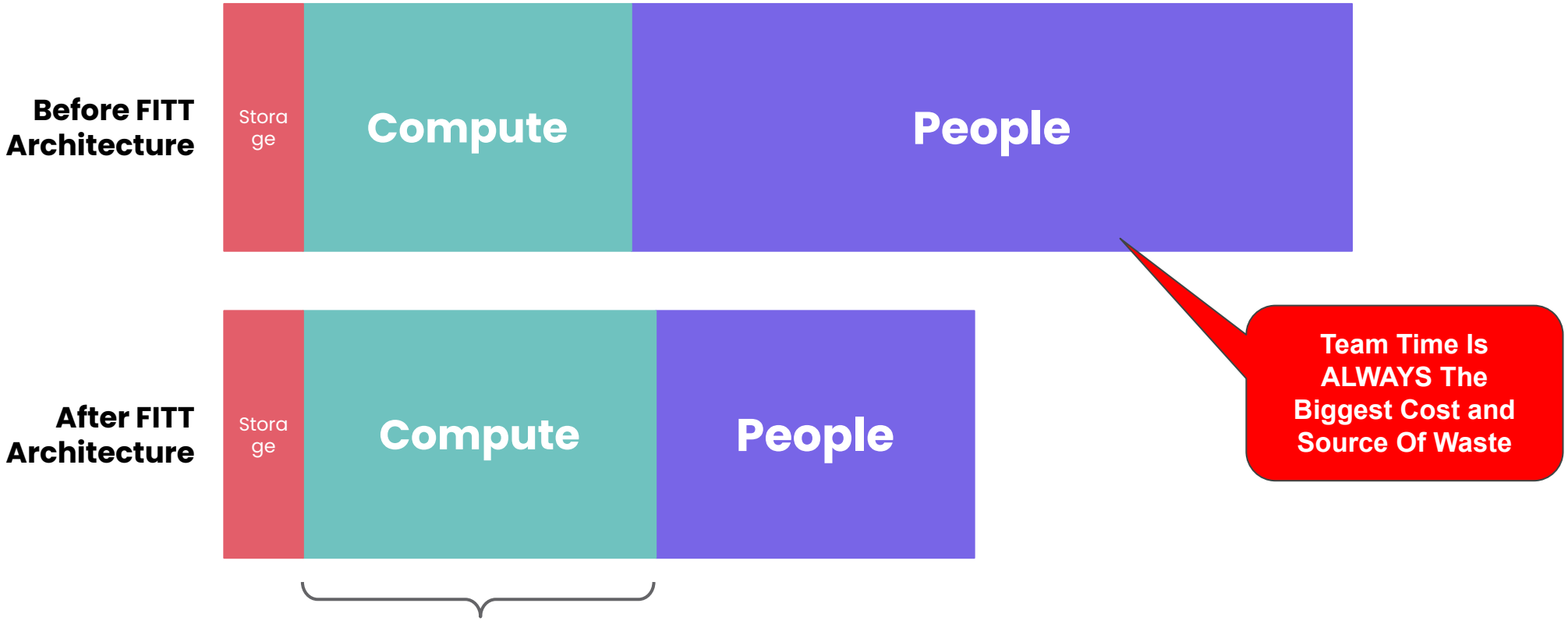
But WAP is not inherently functional or idempotent



How To Get FITT: Operational Rules Of Thumb

- Always be able to rebuild completely from raw.
- Build quality in through automated testing at every step of the process.
- Define Done as released to production.
- Love your errors and improve continuously.
- Own the end-to-end pipeline, encompassing both development and production, and multiple technologies (sql, python, notebooks)
- Test coverage on every table and tool is *very* important
- Concentrate on two critical “down” metrics (errors and cycle time) and two “up” metrics (productivity and customer satisfaction) to measure success

Conclusion: Focus On Reducing Your Biggest Costs

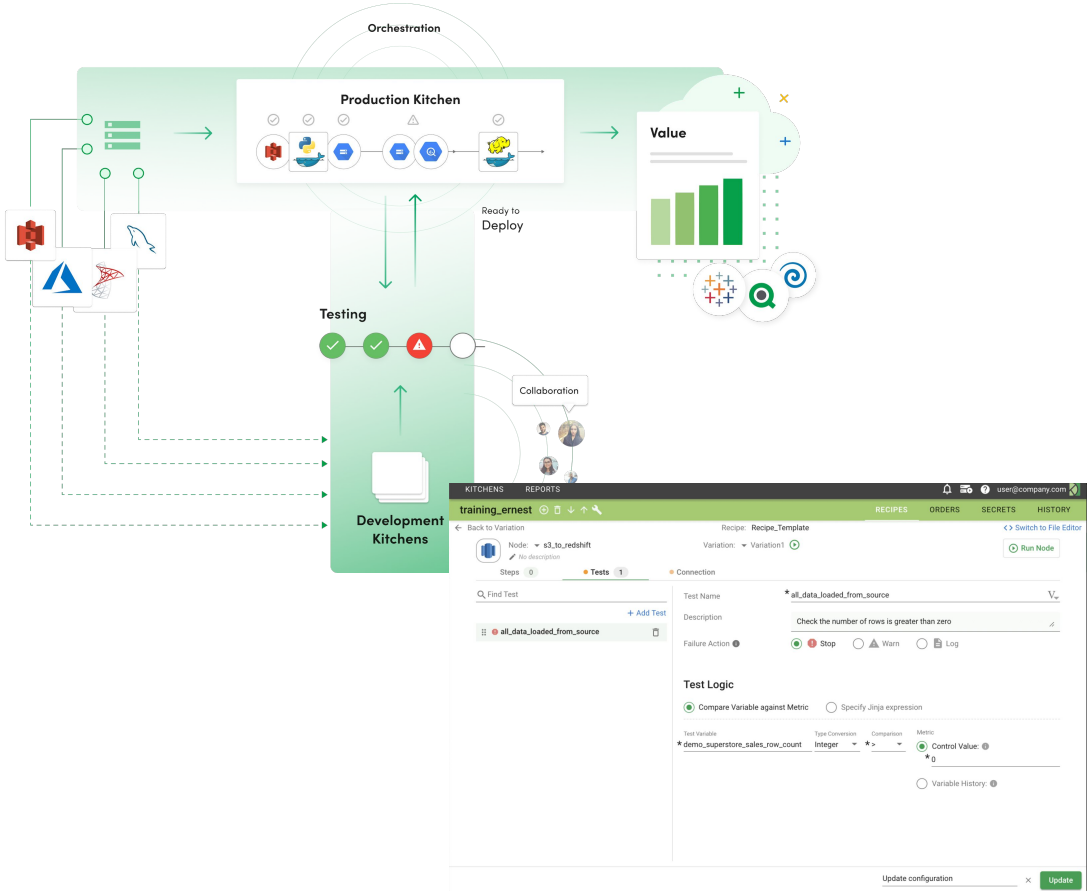


You database and compute cost may go up somewhat. Compute cost is more than offset by reduction (or efficiency increase) from data team productivity

Conclusion (2)

- Data architecture should focus on **maximizing data engineer flow** and **minimize errors**
- FITT isn't magic, but it represents the closest approach we've discovered to making **data engineering "boring" in the best possible sense**—predictable, reliable, and stress-free.
- **No hero-driven development**: If you have to explain how to make a minor fix to a junior data engineer, your data architecture is the problem.
- Senior engineers appreciate the **reduced cognitive load**, allowing them to focus on high-value coding rather than constantly firefighting.

[AD] DataKitchen DataOps Automation Software:
Built For The FITT Data Architecture



DataOps Automation Software

- Build To Run The FITT Data Architecture
- Built To Follow DataOps Principles
- Makes Your Data Engineers 10x Faster

It Does Five Tasks:

1. SQL based ELT
2. Environment Creation and Management
3. Deployment Automation
4. Orchestrate Production Pipelines
5. Testing

DataKitchen

Background Reading

Background on **Functional Programming in Data**

- https://www.dataengineeringweekly.com/p/functional-data-engineering-a-blueprint
- https://maximebeauchemin.medium.com/functional-data-engineering-a-modern-paradigm-for-batch-data-processing-2327ec32c42a
- https://github.com/sbalnojan/easy-functional-data-engineering

Background on **Idempotency In Data**

- https://medium.com/geekculture/idempotent-data-pipeline-ba4c962d8d8c
- https://www.youtube.com/watch?v=uev_27z3-1s

Background on **Testing and Test Coverage in Data**

- https://datakitchen.io/scaling-data-reliability-the-definitive-guide-to-test-coverage/
- https://info.datakitchen.io/data-quality-training-and-certifications

More on **FITT**:

- https://datakitchen.io/fitt-data-architecture/
- https://datakitchen.io/fitt-vs-fragile-sql-orchestration-techniques/

Learn More About DataOps & Data Quality & Data Observability

Install Open Source TestGen
https://info.datakitchen.io/testgen

Install Open Source DataOps Observability
https://docs.datakitchen.io/articles/#!open-source-data-observability/install-data-observability-products-open-source

Sign The DataOps Manifesto
http://dataopsmanifesto.org

Free DataOps Cookbook
https://datakitchen.io/the-dataops-cookbook/

Free DataOps Certification
https://info.datakitchen.io/training-certification-dataops-fundamentals

Free Data Quality & Observability Certification
https://info.datakitchen.io/data-observability-and-data-quality-testing-certification

DataKitchen