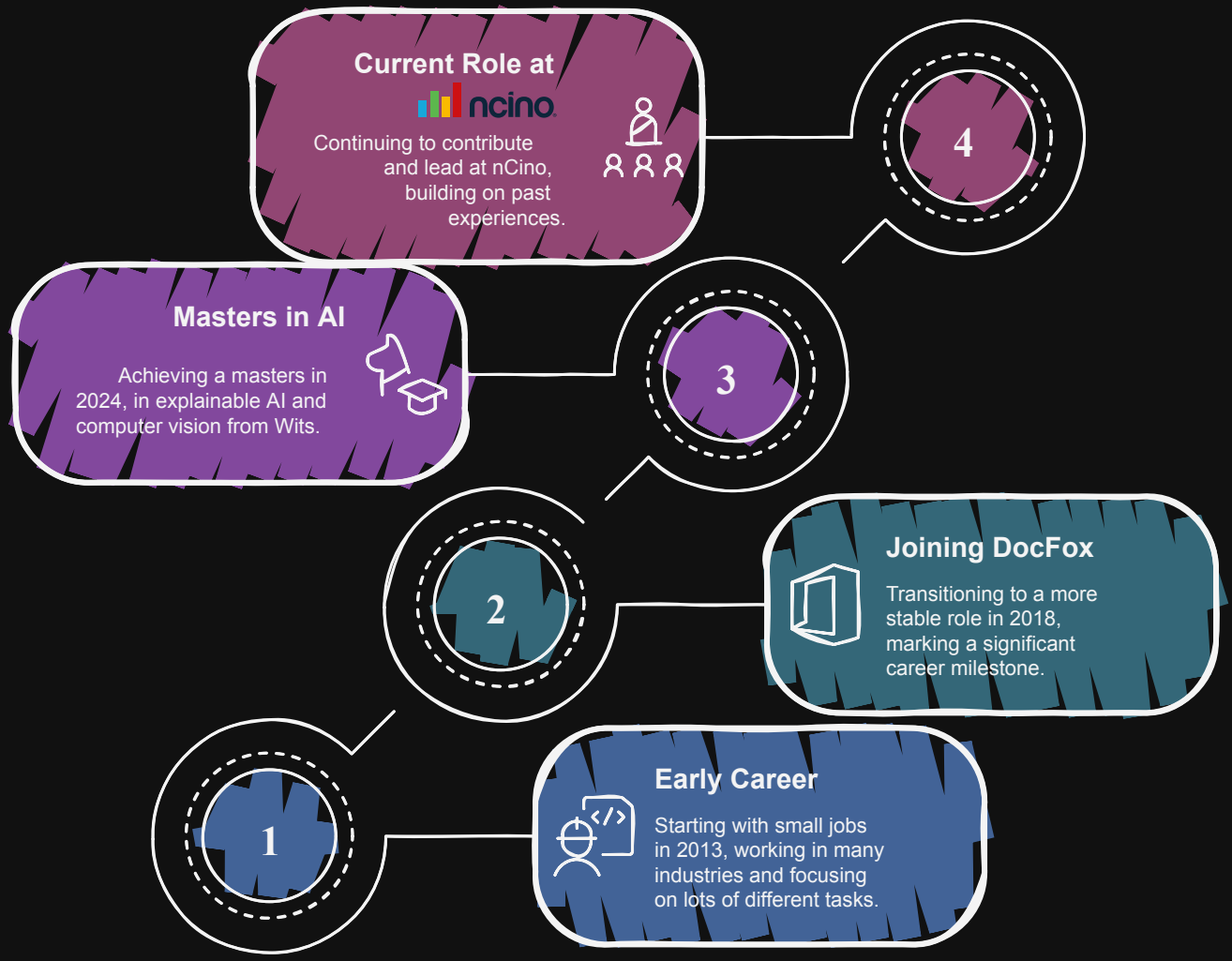
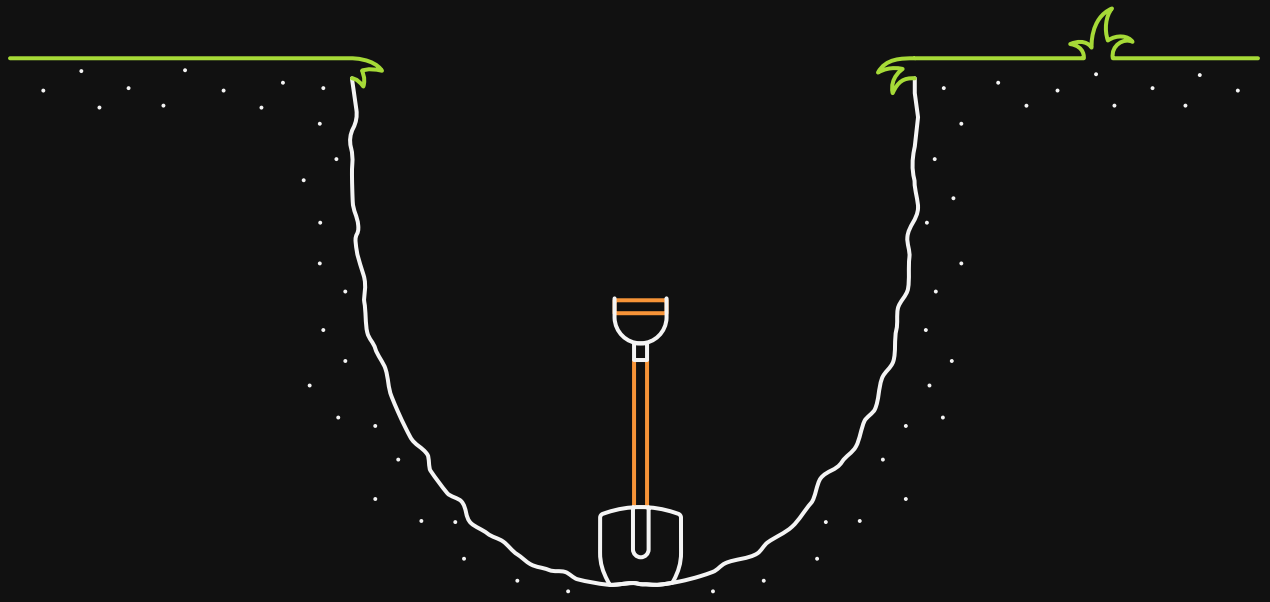
{Building out a sustainable and production ready AI software solution}

From my perspective

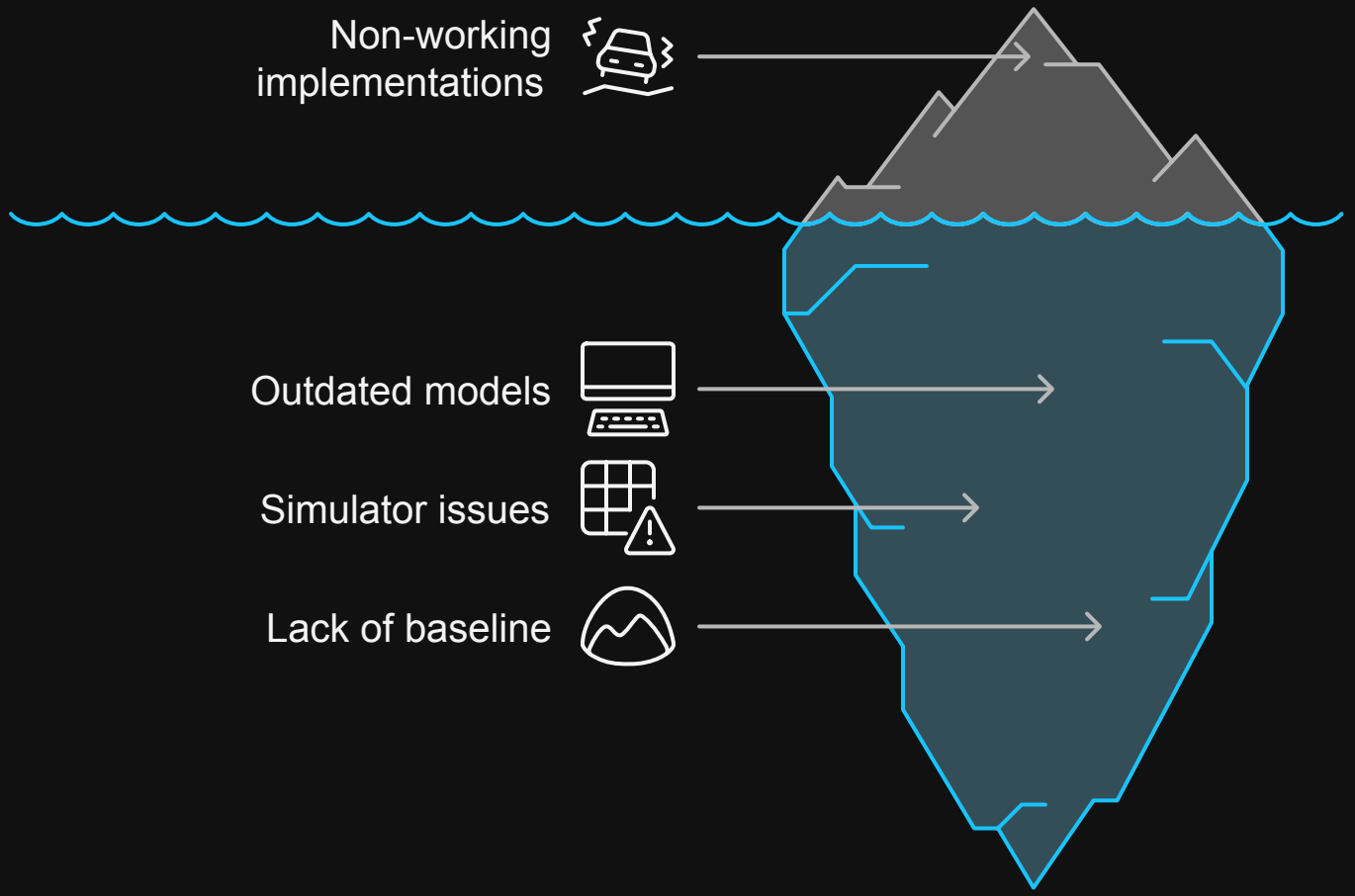
Jason Chalom


Current Role at
ncino
Continuing to contribute and lead at nCino, building on past experiences.
4
Masters in AI
Achieving a masters in 2024, in explainable AI and computer vision from Wits.
3
2
Joining DocFox
Transitioning to a more stable role in 2018, marking a significant career milestone.
1
Early Career
Starting with small jobs in 2013, working in many industries and focusing on lots of different tasks.

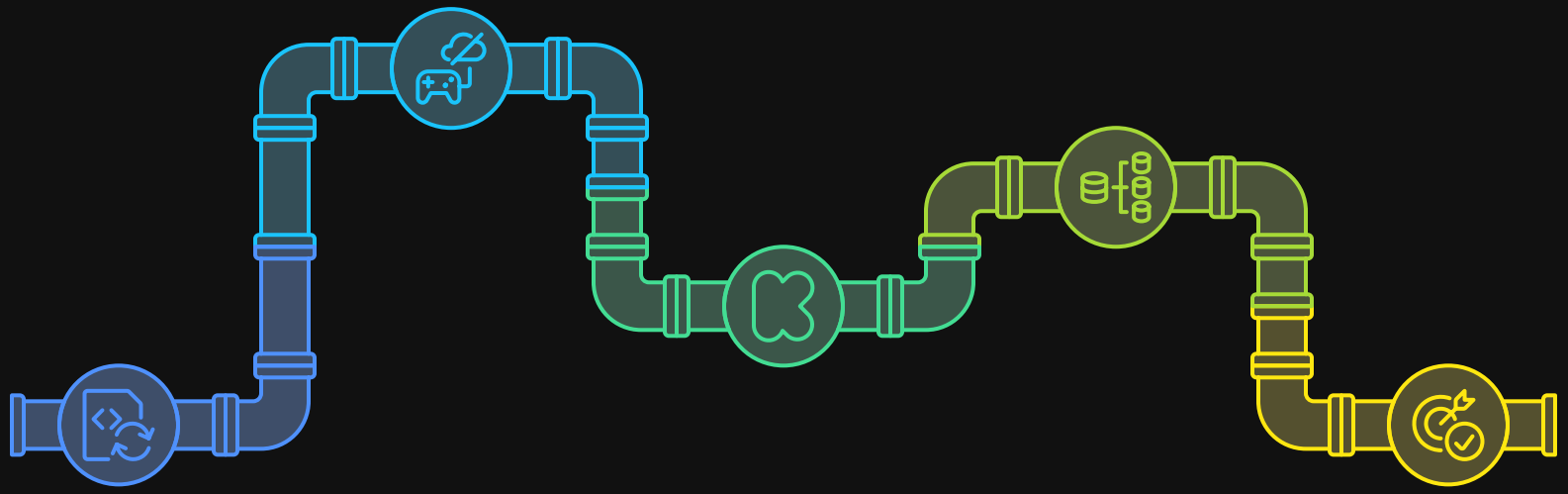
Conflicting views of professional systems vs research



Self-driving car development challenges.



Initial Masters Development Sequence



01

Re-work Code in PyTorch

The initial step involves rewriting the code using PyTorch.

02

Put Simulators on Hold

The decision to temporarily halt simulator development is made.

03

Start Experimental Model Runs

The focus shifts to conducting experimental runs with the models.

04

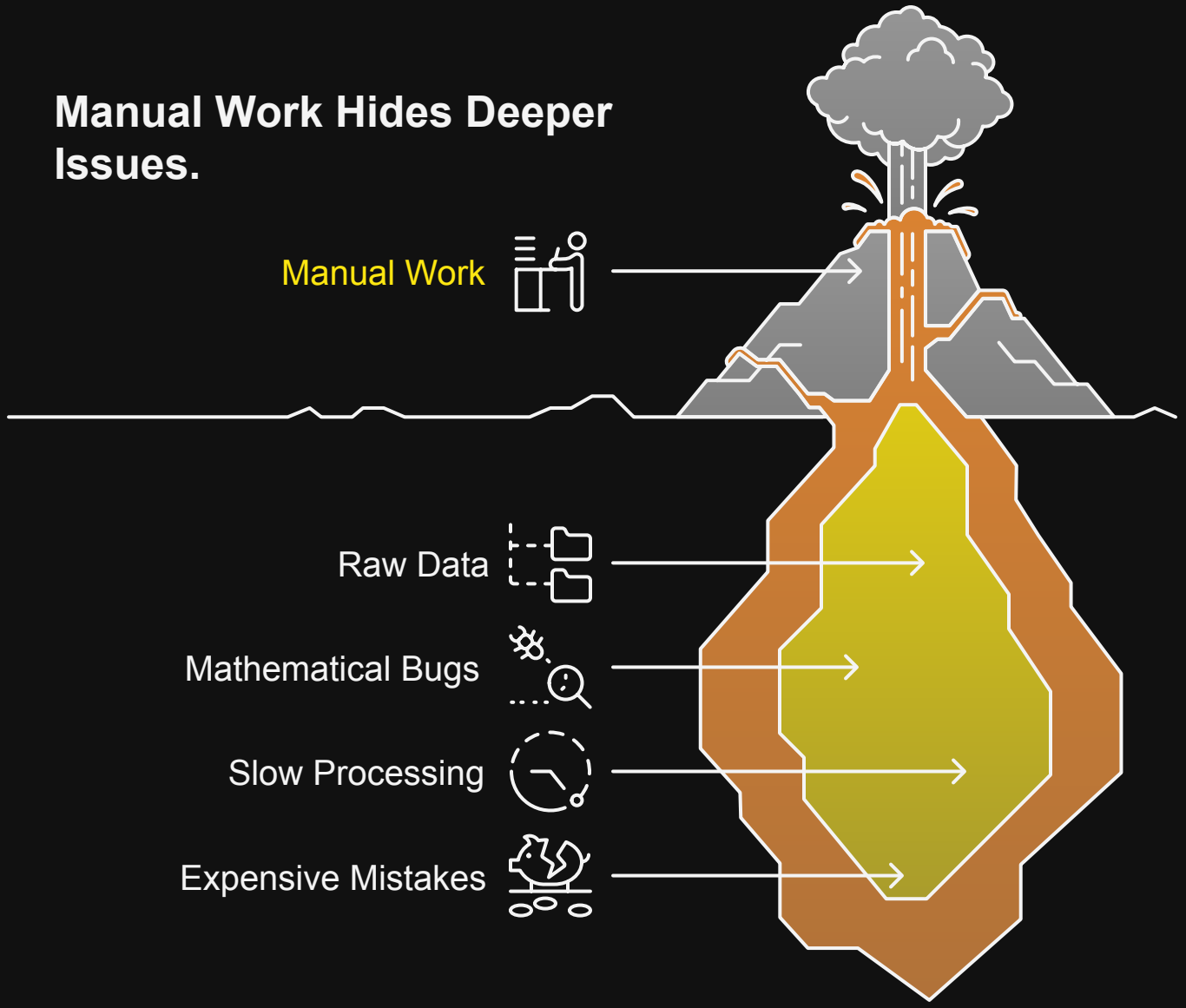
Build Multiple Model Architectures

Various model architectures are developed to explore different approaches.

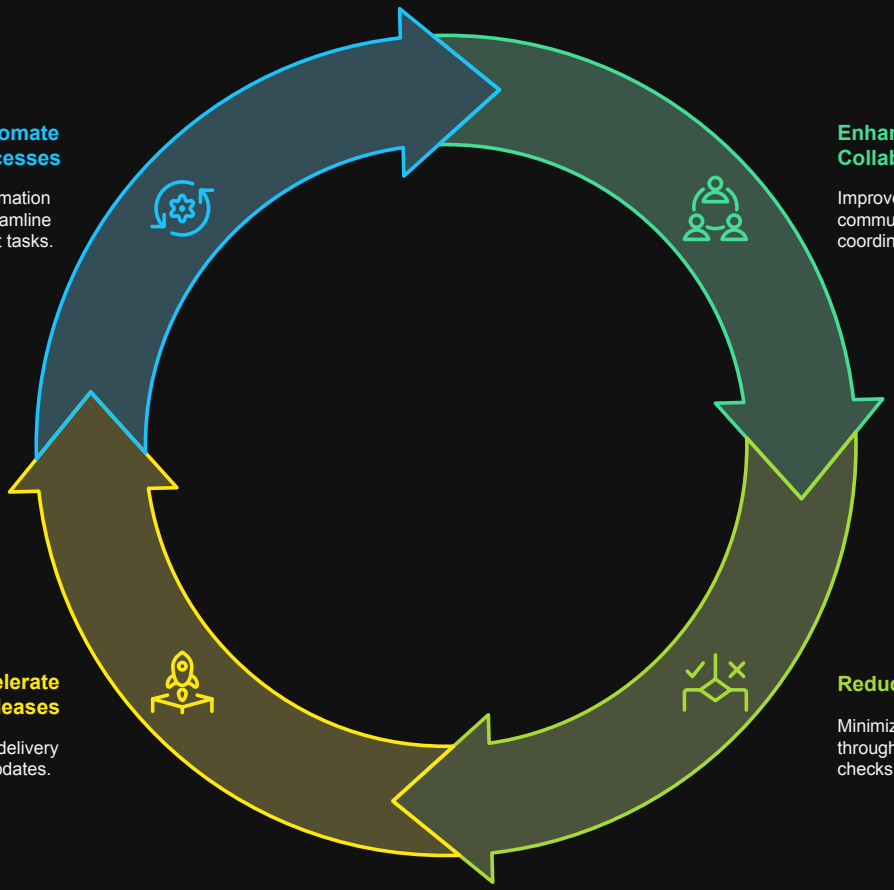
05

Get a Result as Fast as Possible

The goal is to achieve a result quickly to assess progress.


Manual Work Hides Deeper Issues.
Manual Work
Raw Data
Mathematical Bugs
Slow Processing
Expensive Mistakes

CI/CD Cycle



Automate Processes

Implement automation tools to streamline development tasks.

Enhance Collaboration

Improve team communication and coordination.

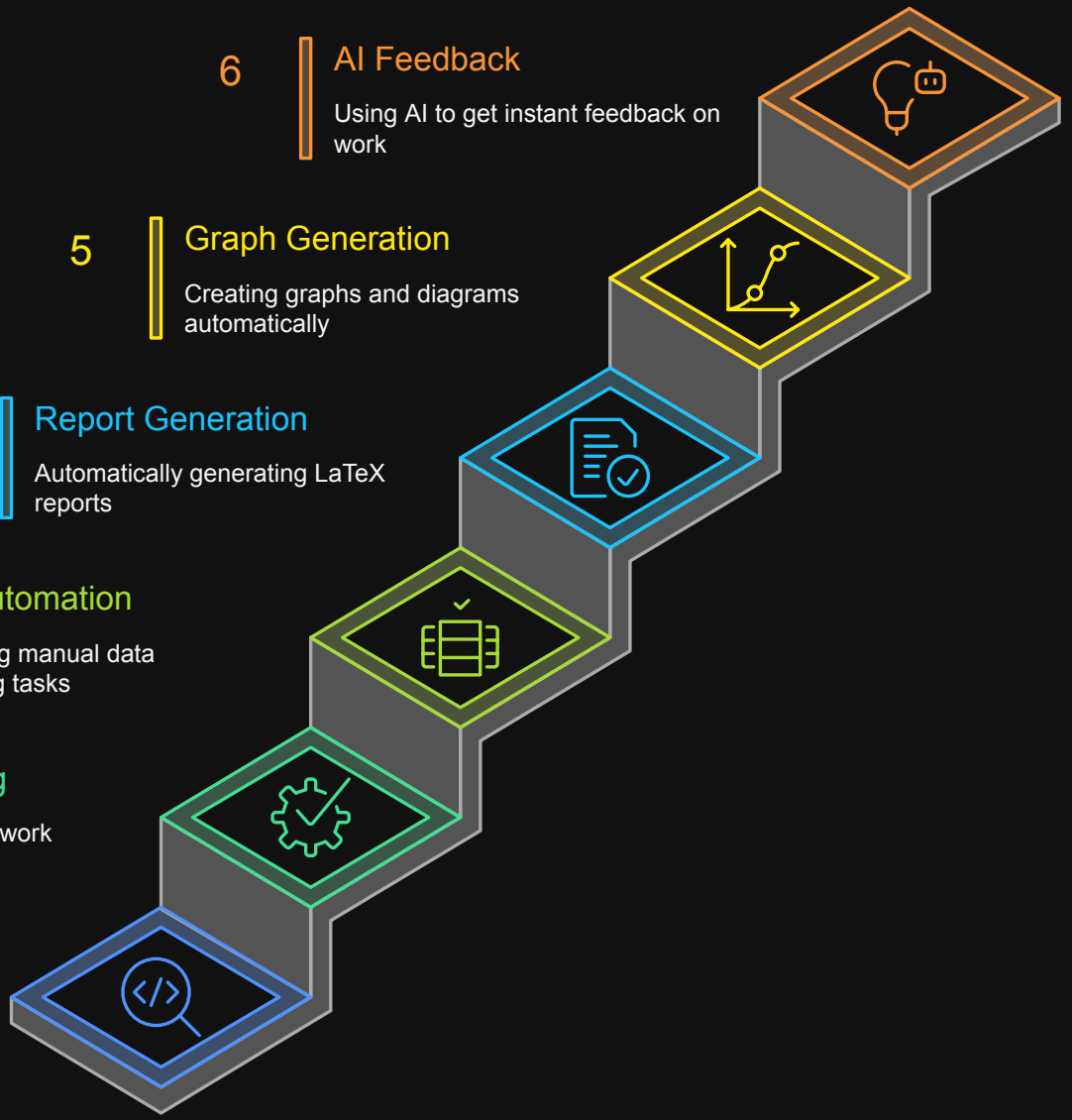
Accelerate Releases

Speed up the delivery of software updates.

Reduce Errors

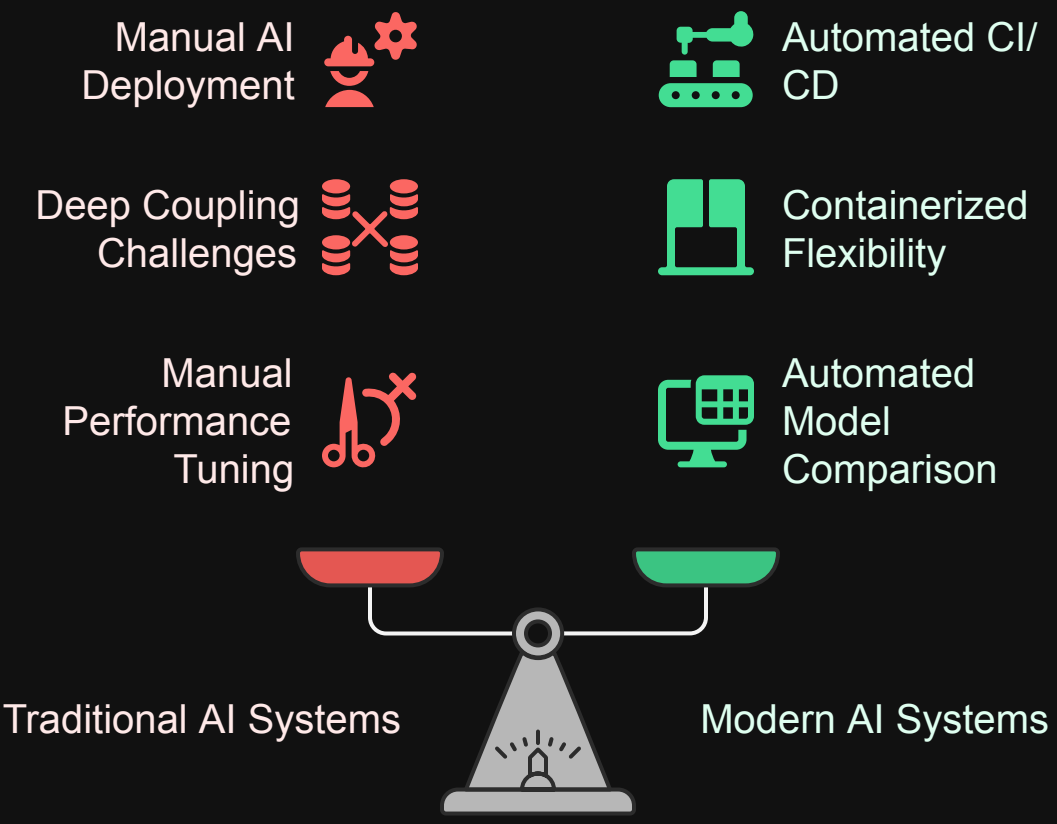
Minimize mistakes through automated checks and tests.

Achieving Comprehensive Workflow Automation



1. **Unit Testing**
   Testing individual components for correctness
2. **Integration Testing**
   Ensuring different parts work together seamlessly
3. **Data Automation**
   Automating manual data processing tasks
4. **Report Generation**
   Automatically generating LaTeX reports
5. **Graph Generation**
   Creating graphs and diagrams automatically
6. **AI Feedback**
   Using AI to get instant feedback on work

Transitioning from Manual to Automated AI Systems



AI Development Practices I think Work

Assertions

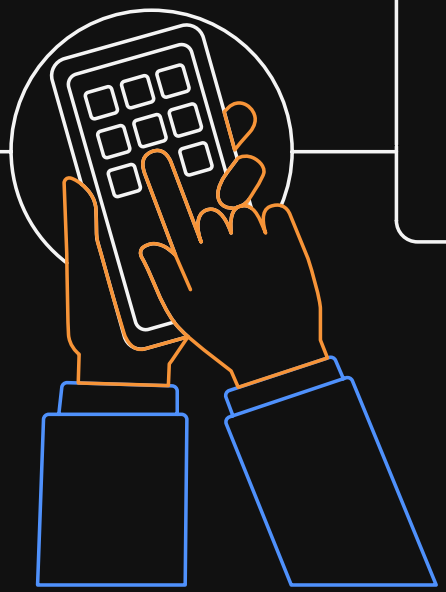
Identifying known good states and embedding those as hard escape conditions.

Model Agnostic Implementations

Easily replaceable components.

Dynamic Testing

Testing over time, checking for data drift, data removal changing behaviour and new cases.



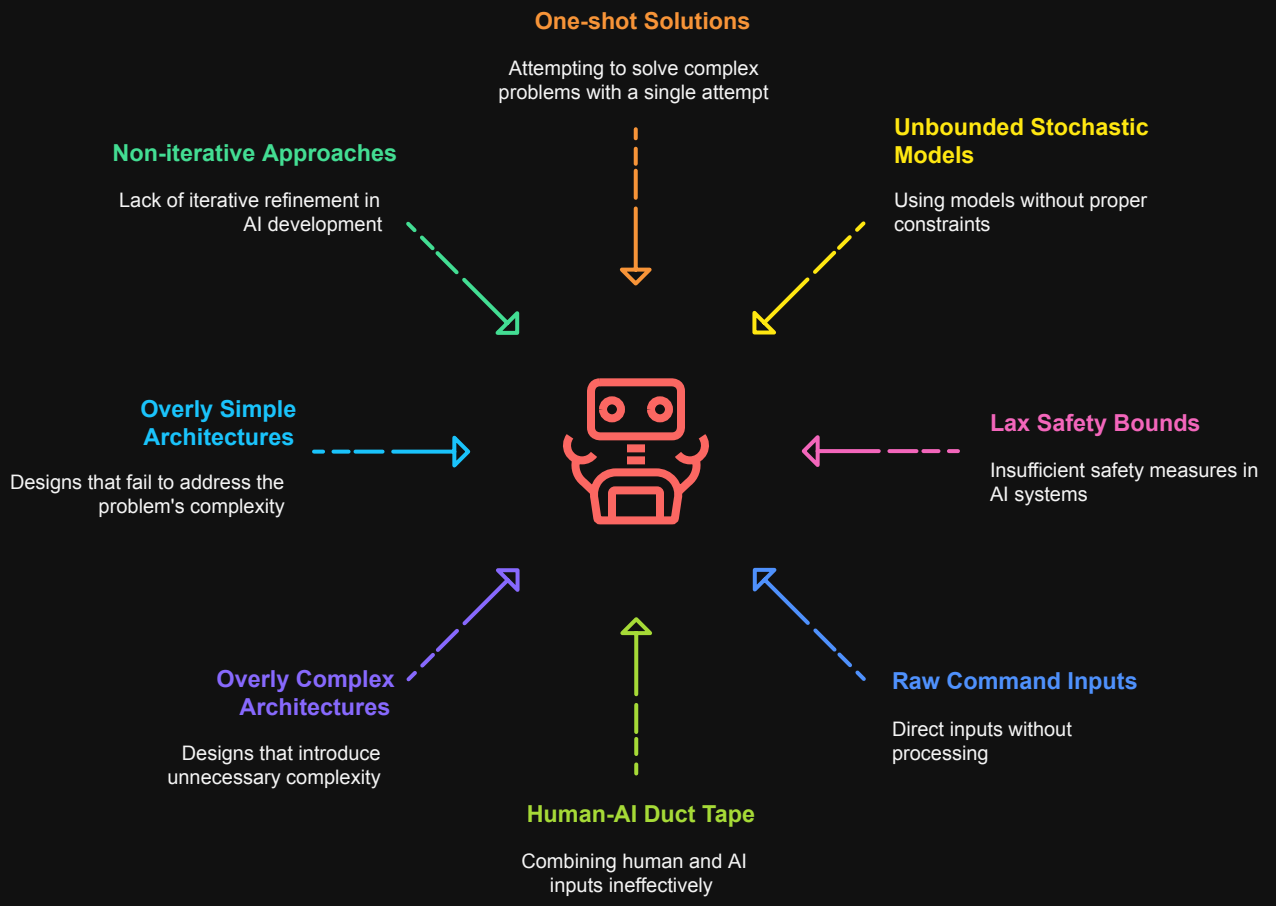
Monitoring

Bounded / expected behaviour, simple parts like ram, memory and other cost factors, failure and error rates.

Testing

Exercising code to handle happy and sad paths.

Common Pitfalls in AI System Design



What comes next?

Putting Ideas Into Action


Lila Cafe

The problem of reciepts

1.

Many purchases

We want to have detailed data of stock prices and volumes

2.

Scanning is easy

We already scan all documents and receipts

3.

Creating spreadsheets is time consuming

Manually copying out each item is difficult

** WELCOME TO **
ENGEN C&B MOTORS
VAT: 4540318385

QS 3KG ICE2 17.00
QS 3KG ICE2 17.00

ITEMS 2 TOTAL 34.00

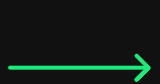
EFT-Retail 34.00
485442*******1683
CHANGE 0.00

Join MySchool and Support Your Community

VAT-CODE NET-VAL VAT-VAL
STANDARD RATE/S 15.0% 29.56 4.44

MA 01-00325-5557 07:50:40 07/06/2025

THANK YOU FOR SHOPPING AT
ENGEN C&B MOTORS



	A	B	C	D	E	F	G
1	Item Name	Store	VAT Number	Date	Price	VAT	Reciept ID
2	QS 3KG ICE2	ENGEN C&B MOTORS	4540318385	07:50:40 07/06/2025	R17.00	R4.44	1
3	QS 3KG ICE2	ENGEN C&B MOTORS	4540318385	07:50:40 07/06/2025	R17.00	R4.44	1

A simple script

```
def process_receipt(image_path):
  model_name = "naver-clova-ix/donut-base-finetuned-cord-v2"
  cache_dir = "./model_cache"

  processor = DonutProcessor.from_pretrained(model_name, cache_dir=cache_dir)
  model = VisionEncoderDecoderModel.from_pretrained(model_name, cache_dir=cache_dir)

  device = "cpu"
  model.to(device)

  image = Image.open(image_path).convert('RGB')

  task_prompt = "<s_cord-v2>"
  decoder_input_ids = processor.tokenizer(task_prompt, add_special_tokens=False, \
    return_tensors="pt").input_ids

  pixel_values = processor(image, return_tensors="pt").pixel_values

  outputs = model.generate(
    pixel_values.to(device),
    decoder_input_ids=decoder_input_ids.to(device),
    max_length=model.decoder.config.max_position_embeddings,
    pad_token_id=processor.tokenizer.pad_token_id,
    eos_token_id=processor.tokenizer.eos_token_id,
    use_cache=True,
    bad_words_ids=[[processor.tokenizer.unk_token_id]],
    return_dict_in_generate=True,
  )

  sequence = processor.batch_decode(outputs.sequences)[0]
  sequence = sequence.replace(processor.tokenizer.eos_token, "") \
    .replace(processor.tokenizer.pad_token, "")
  sequence = sequence.replace(task_prompt, "")

  return sequence
```

A simple script

```
OCR Result:
<s_menu><s_nm> WELCOME TO</s_nm><s_unitprice> ENGEN C&B MOTORS</s_nm><s_unitprice>
4540318385</s_unitprice><s_cnt> QS 3KG ICE2</s_nm><s_unitprice> 17.00</s_unitprice><s_cnt> QS SKG
ICE2</s_nm><s_price> 17.00</s_price></s_menu><s_sub_total><s_subtotal_price> 34.00</s_subtotal_price>
</s_sub_total><s_total><s_total_price> 34.00</s_total_price><s_changeprice> 0.00</s_changeprice>
<s_creditcardprice> 48542</s_unitprice><s_cnt> *1683</s_creditcardprice><s_menuqty_cnt> 16内
</s_menuqty_cnt></s_total>
```

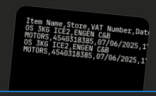
Receipt Data CSV Extraction

J Extract out information from the given receipt as csv with these headings

```
Item Name,Store,VAT Number,Date,Price,VAT,Reciept ID
```

Looking at the receipt, I can extract the following information:

Receipt Data CSV
Code



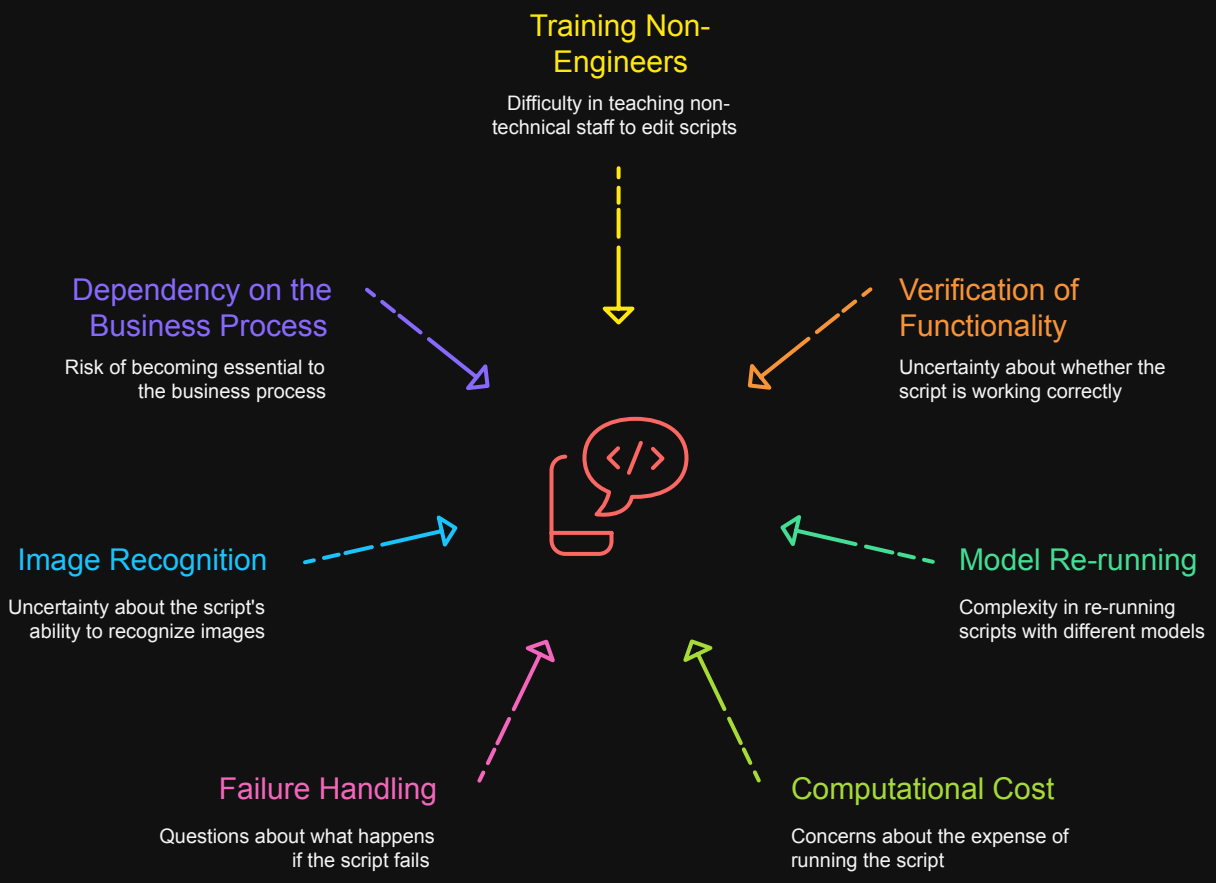
I've extracted the receipt information into CSV format. The receipt shows:

- Two items of "OS 3KG ICE2" at R17.00 each
- Store: ENGEN C&B MOTORS
- VAT Number: 4540318385
- Date: 07/06/2025 (based on the timestamp format)
- Total VAT: R4.44 (split equally between the two items)

Copy | Publish

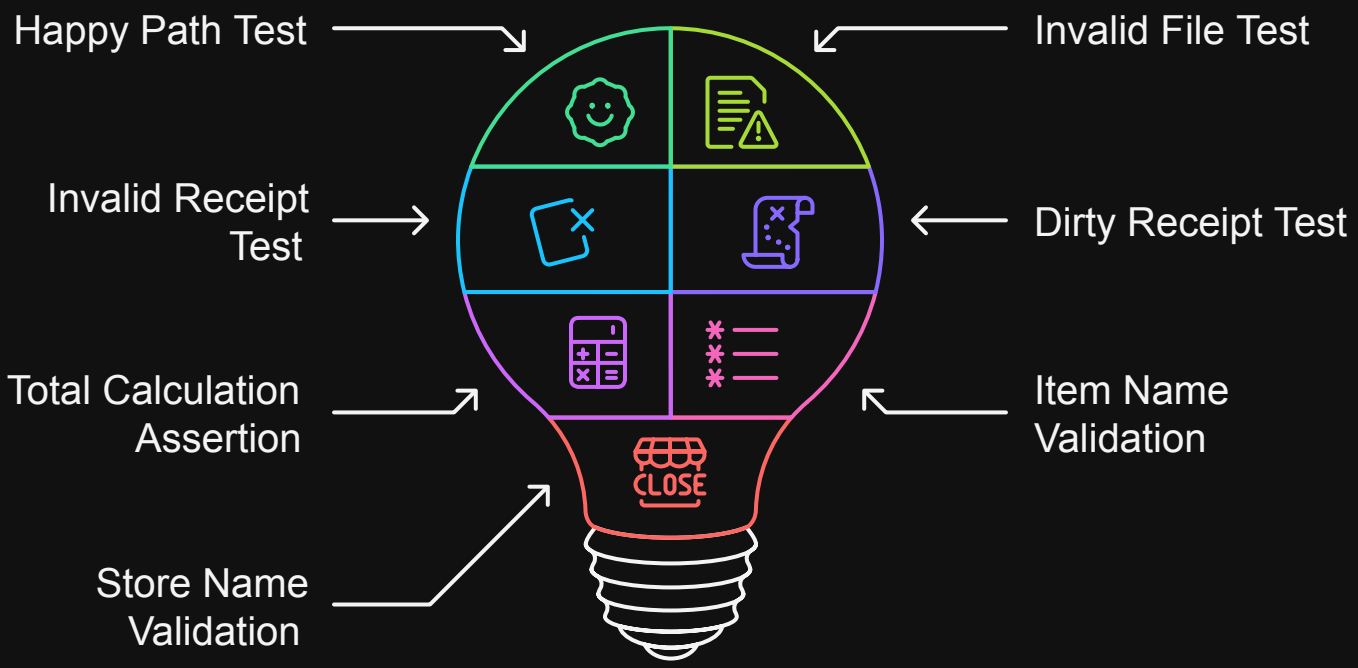
```
Item Name,Store,VAT Number,Date,Price,VAT,Receipt ID
OS 3KG ICE2,ENGEN C&B MOTORS,4540318385,07/06/2025,17.00,4.44,01-00325-5557
OS 3KG ICE2,ENGEN C&B MOTORS,4540318385,07/06/2025,17.00,4.44,01-00325-5557
```

Challenges in Script-Based Solutions

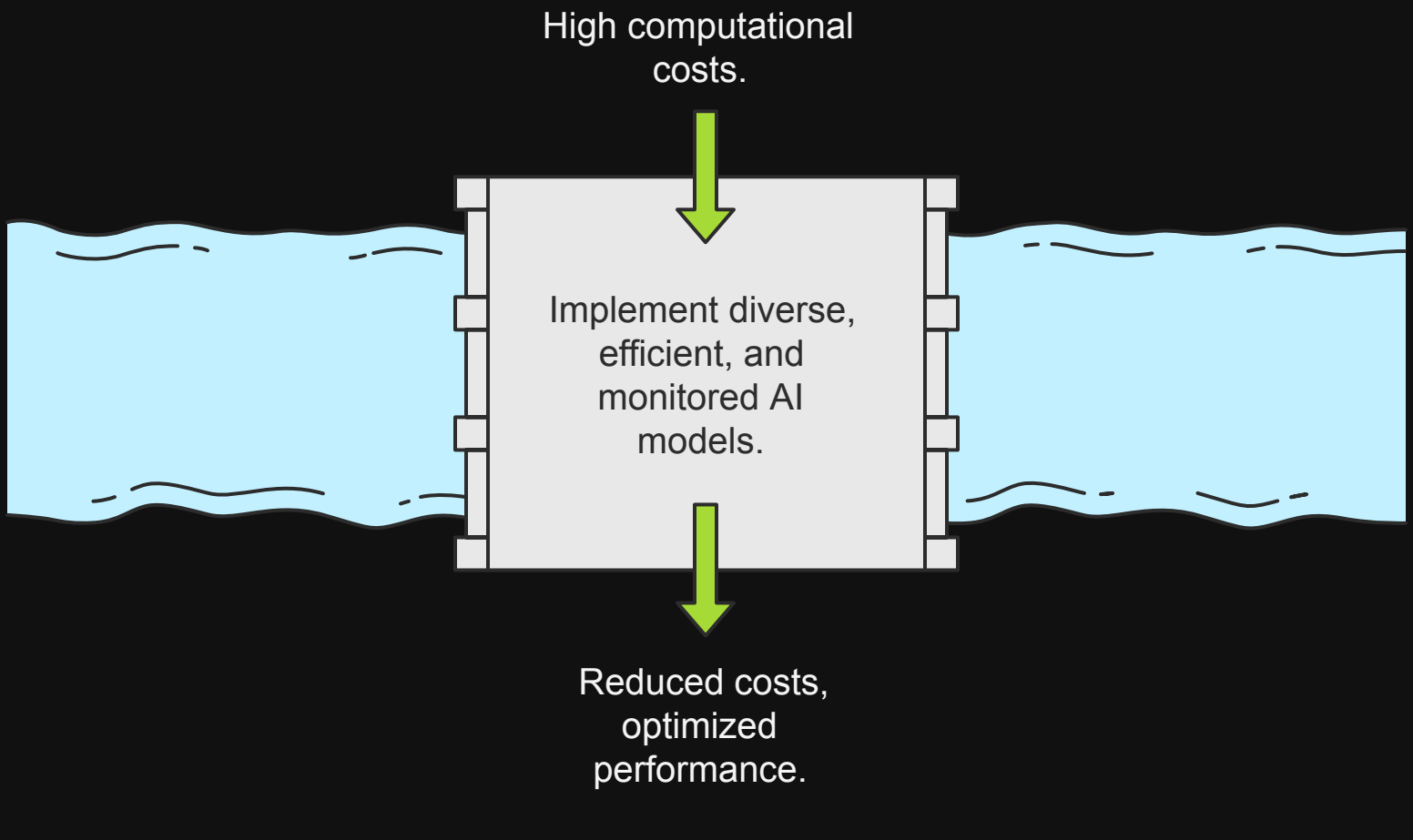


Task Scheduler Pattern Cycle

Enhancing System Reliability



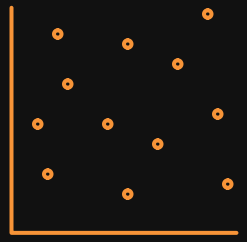
Improving Coffee Shop Sustainability with Efficient Models



Monitoring aspects

Simple Logs

Monitoring logs for errors and unusual activity.



Data Drift

Repeating analysis to detect changes in data patterns.

Alarms and Alerts

Setting up alarms and alerts for critical issues.

Thank you

contact@jasonchalom.com

AI Tools Used:

- Claude.ai
- Napkin.ai
- Mistral

Other Tools Used:

- slides.com
- Sublime Text
- Inkscape
- GIMP

Thank you

contact@jasonchalom.com

My GitHub

My Blog