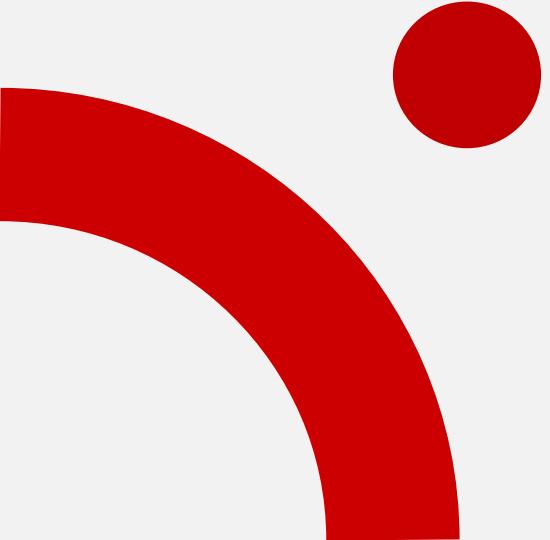


# Innovating Laziness

Greg Case

he/him



# **Today's Discussion**

A Quick History in Laziness

Value of Automation

Benefits of Automation

Examples/Demos

# History of Laziness

## 3600 BC - Writing

Before the invention of writing, information was shared via word of mouth. The invention of writing was an early form of automation!  
Write once, read anywhere!



<https://www.flickr.com/photos/charlestilford/2552654321/>

# History of Laziness

## 1000 BC – The Water Wheel

The water wheel was used to grind grain into flour, freeing up human labor for other tasks.



[https://commons.wikimedia.org/wiki/File:Kirk\\_Mills\\_water\\_wheel.JPG](https://commons.wikimedia.org/wiki/File:Kirk_Mills_water_wheel.JPG)

# History of Laziness

## 1436 – The Printing Press

The printing press allowed the mass production of writing. Invented by Johannes Gutenberg, it was initially used to mass produce 200 copies of the bible, taking 3 years to do so.



# History of Laziness

## 18<sup>th</sup> Century The Industrial Revolution

The Industrial Revolution was marked by using machinery to assist in manufacturing. Aided by the invention of the steam engine, it led the mass production of goods, interchangeable parts, and impacted nearly all facets of society.



# History of Laziness

## 1804 - Jacquard Loom

The Jacquard Loom allowed looms to weave complicated patterns, instructed by a set of punch cards that would instruct the loom.



*By Stephencdickson - Own work, CC BY-SA 4.0,  
<https://commons.wikimedia.org/w/index.php?curid=79746138>*

# History of Laziness

## 1837 – The Analytical Engine

Charles Babbage designed the analytical engine, capable of arithmetic, control flow, and had a working memory. Although never fully built, it paved the way for modern computing. It was controlled by punch cards.

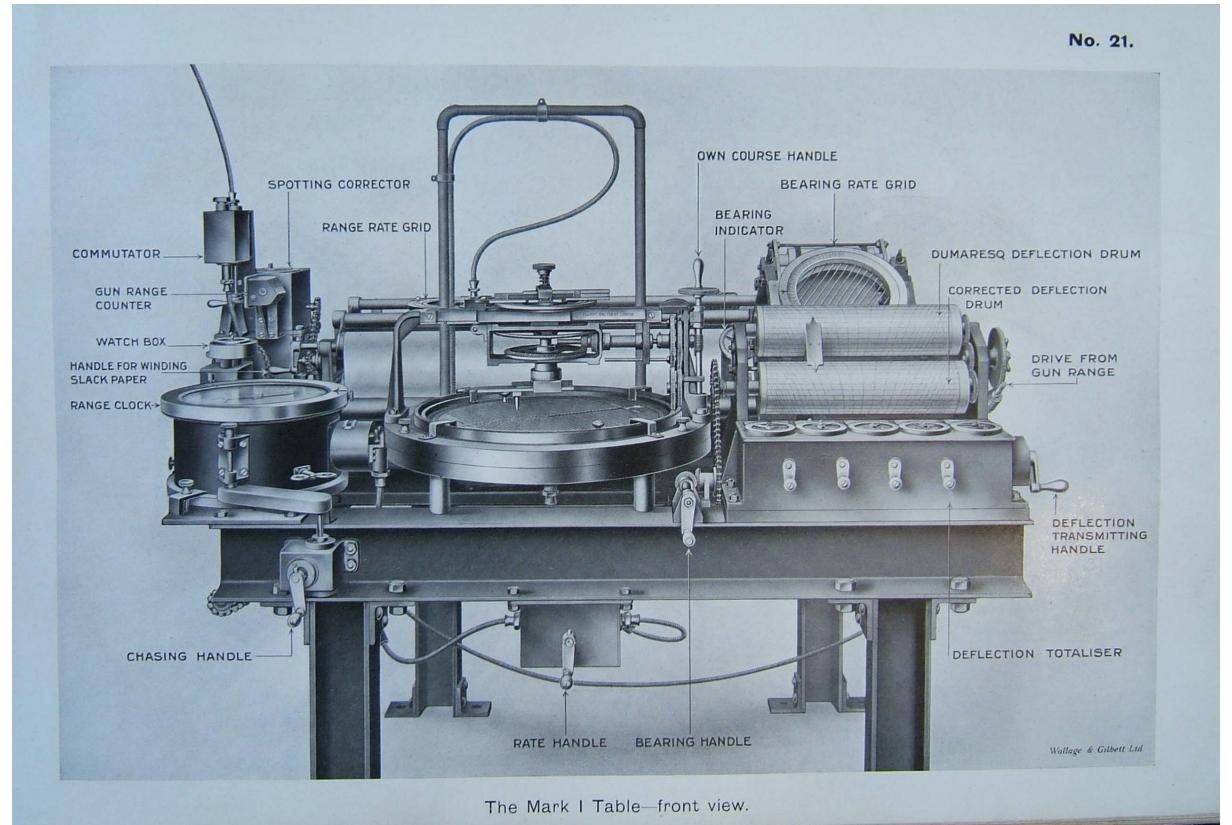


*By Marcin Wichary from San Francisco, U.S.A. - Analytical Engine Mill, CC BY 2.0,  
<https://commons.wikimedia.org/w/index.php?curid=3614794>*

# History of Laziness

## 1911 – The Dreyer Fire Control Table

An early mechanical computer used in World War I to calculate Range and Bearing Plots between ships at sea, utilizing a circular device called a Dumaresq to calculate the vectors.



<http://www.dreadnoughtproject.org/>

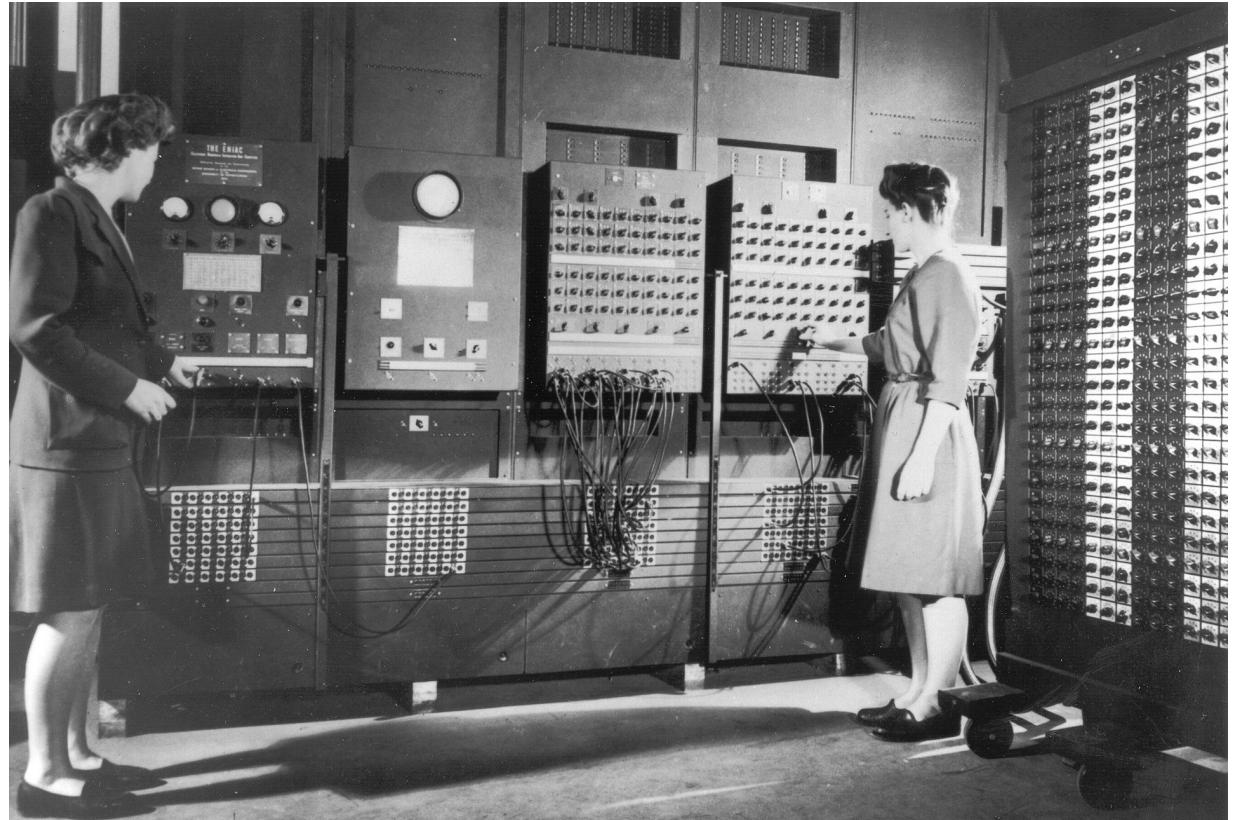
# History of Laziness

1945 – ENIAC

**Electronic Numerical Integrator And Computer**

The first electronic, programmable, general purpose computer.

- 18,000 Vacuum Tubes
- 7,200 Crystal Diodes
- 70,000 Resistors
- 5,000,000 Hand Soldered Connections
- 27 Tons

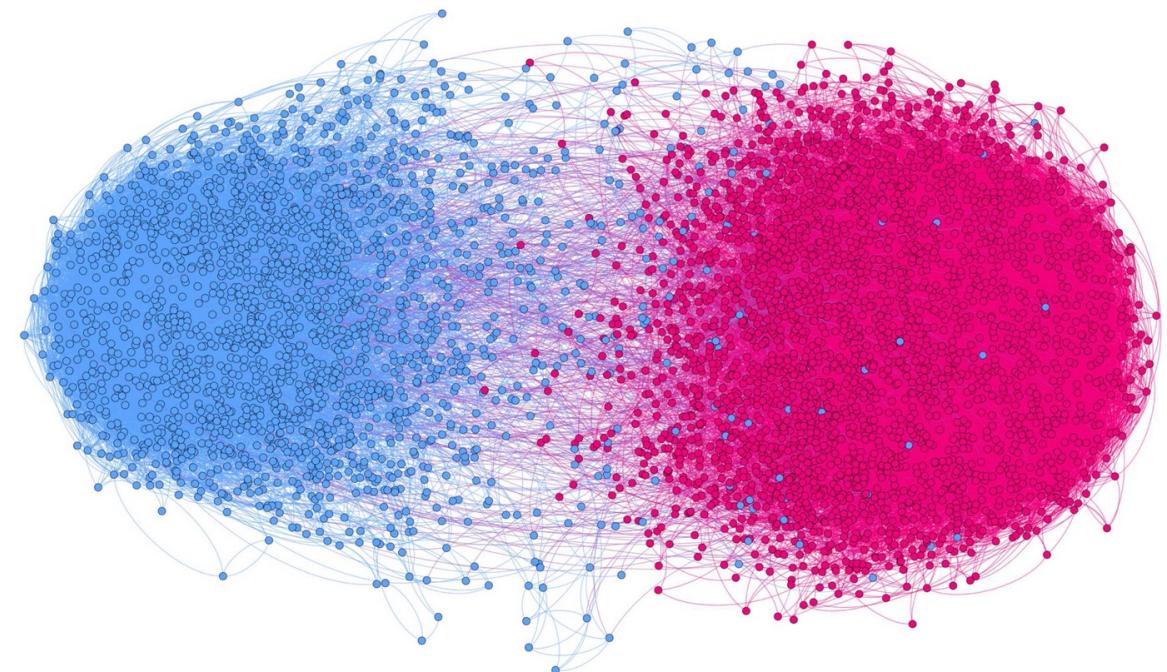


<https://en.wikipedia.org/wiki/ENIAC>

# History of Laziness

## Present Day – Modern Computing

Someone having a bad day can now enrage 1000s of people around the globe with a single tweet, all while sitting on their couch!



Example of a polarized and segregated network on Twitter

<https://link.springer.com/article/10.1007/s42001-020-00084-7/figures/1>

# You are a superhero



*Generated by DALL-E*

*"a impressionist oil painting of a smiling female super hero programming a computer"*

# AI - Automation Force Multiplier

---

AI is a tool in my toolbox that can help me understand the other tools in my toolbox.

LLM do not “know” anything. They are very good at generating things that, statistically speaking, more correct than the gibberish.



**joyce carol manor**

@jon-christian.bsky.social

computer scientists: we have invented a virtual dumbass who is constantly wrong

tech CEOs: let's add it to every product

May 31, 2024 at 5:27 PM

---

**1.7K** reposts **5.6K** likes

---

<https://bsky.app/profile/jon-christian.bsky.social/post/3ktsxyw2pf423>

# Value of Automation

HOW LONG CAN YOU WORK ON MAKING A ROUTINE TASK MORE  
EFFICIENT BEFORE YOU'RE SPENDING MORE TIME THAN YOU SAVE?  
(ACROSS FIVE YEARS)

		HOW OFTEN YOU DO THE TASK					
		50/DAY	5/DAY	DAILY	WEEKLY	MONTHLY	YEARLY
1 SECOND		1 DAY	2 HOURS	30 MINUTES	4 MINUTES	1 MINUTE	5 SECONDS
5 SECONDS		5 DAYS	12 HOURS	2 HOURS	21 MINUTES	5 MINUTES	25 SECONDS
30 SECONDS		4 WEEKS	3 DAYS	12 HOURS	2 HOURS	30 MINUTES	2 MINUTES
HOW MUCH TIME YOU SHAVE OFF	1 MINUTE	8 WEEKS	6 DAYS	1 DAY	4 HOURS	1 HOUR	5 MINUTES
	5 MINUTES	9 MONTHS	4 WEEKS	6 DAYS	21 HOURS	5 HOURS	25 MINUTES
	30 MINUTES	6 MONTHS	5 WEEKS	5 DAYS	1 DAY	2 HOURS	
	1 HOUR	10 MONTHS	2 MONTHS	10 DAYS	2 DAYS	5 HOURS	
	6 HOURS			2 MONTHS	2 WEEKS	1 DAY	
1 DAY					8 WEEKS	5 DAYS	

Source: <https://xkcd.com/1205>

$$S = (T * R) - A$$

S – Savings

T – Time Spent on task

R – Repetitions of task

A – Time spent automating task

$$V = C * ((T * R) - A)$$

V – Value

T – Time spent on task

R – Repetitions of task

A – Time spent automating task

C – Opportunity Cost

**THAT'S NOT ENOUGH**

**WE HAVE TO GO DEEPER**

# What is the cost of making a mistake?

## Wasted Time

$$MC = OC * T * TM$$

MC = Mistake Cost

OC = Opportunity Cost

T = Time

TM = Team Members Impacted

## Guest/Operations Impact

Revenue?

Cost of goods?

Cost of labor?

Guest perception?

$$V = [C * ((T * R) - A)] + (MC * MP)$$

V – Value

T – Time Spent on Task

R – Repetitions of Task

A – Time spent automating Task

C – Opportunity Cost

MC – Mistake Cost

MP – Mistake Probability

$$V = [C * ( TM * (T * R) - A ) ] + (MC * MP)$$

V – Value

T – Time Spent on Task

R – Repetitions of Task

A – Time spent automating Task

C – Opportunity Cost

MC – Mistake Cost

MP – Mistake Probability

TM – Team Members

$$V = C * (TM * (T * R)) - A$$
$$\$3573 = \$.03 * (8 * (60 * 250)) - 900$$

V – Value = \$3308.33

C – Opportunity Cost = \$.03

TM – Team Members = 8

T – Time Spent on Task = 60 Seconds

R – Repetitions of Task = 250

A – Time spent automating Task = 900 Seconds

# Example: DPCI / TCIN Lookup

$$V = [C * (TM * (T * R) - A)] + (MC * MP)$$

Variable	Description	
C	Annualized Engineer Value	\$ 200,000.00
R	Opportunity Cost Per Second Per Day	\$ 0.03 1
T	Per Year	250
A	Time Saved Minutes) Time Saved (Seconds)	1 60
T*R	Time To Automate (Minutes)	15
T*R-A	Time To Automate (Seconds)	900
TM	Time Saved Gross	15,000
	Time Saved Net	14,100
	Team Members	8
V	Annualized Value to Company per Engineer	\$ 423.00
	Annualized Value to Company per Team	\$ 3,573.00

# TCIN / DPCI Translation Functions

```
function dpci-to-tcin() {
    local key=█████████████████████████████████████
    local dpci=$1
    local url="https://api-internal.target.com/items/v4/canonical_tcin/$dpci"
    curl -sL -H "x-api-key: $key" "$url" | jq -r '.tcin'
}

function tcin-to-dpci() {
    local key=█████████████████████████████████████
    local tcin=$1
    local url="https://api-internal.target.com/items/v4/graphql/compact/item/?selection=ADEBAAAARhNgoAJQAAA&tcin=$tcin"
    curl -sL -H "x-api-key: $key" "$url" | jq -r '.data.item.alternateIdentifiers.dpci'
}
```

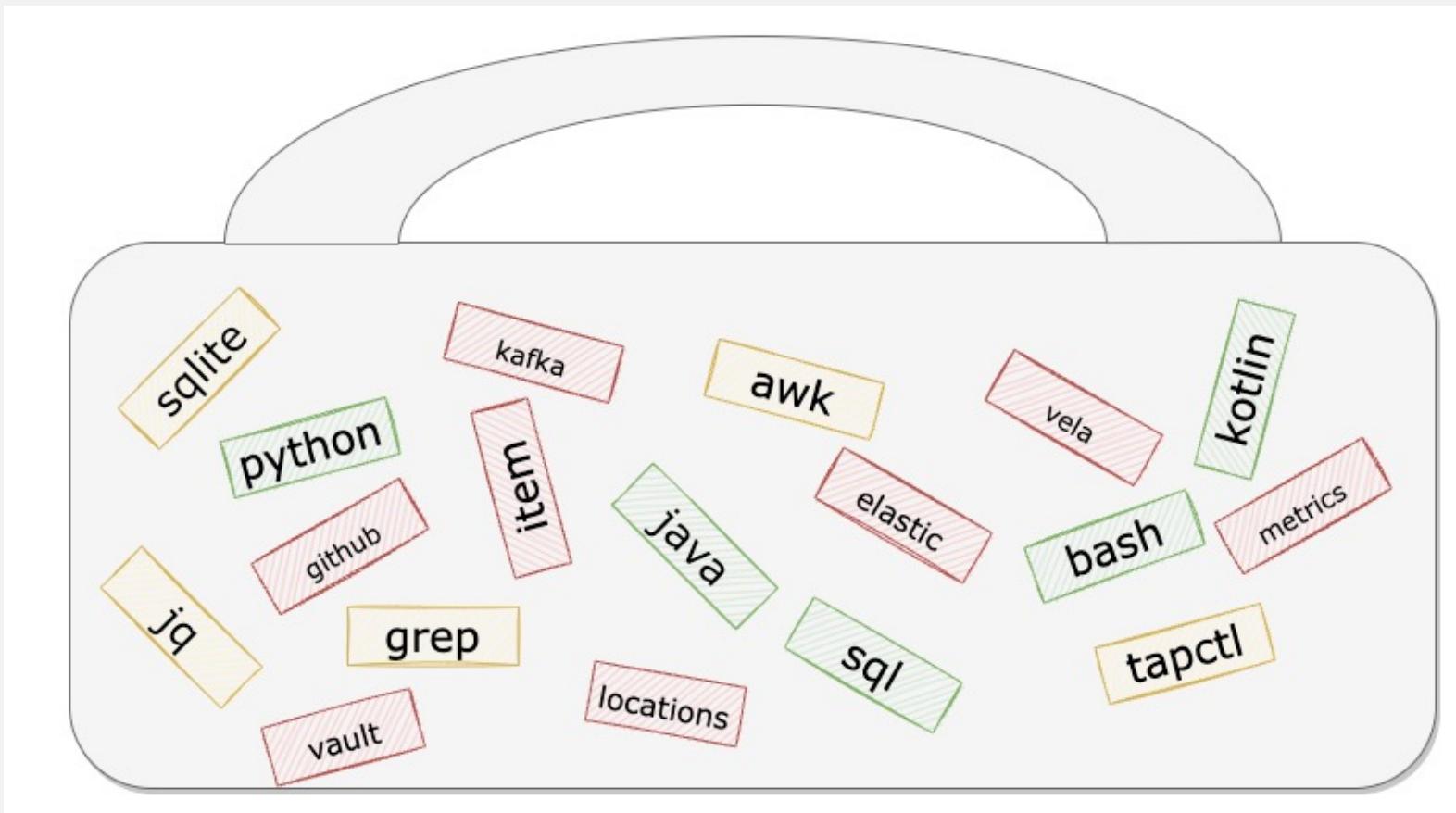
# **Benefits of Automation**

Adds more tools to your toolbox

You will build better software

Team Communication/Knowledge Sharing

# What do you have in your toolbox?



# Example



# Better Software

Discoverability

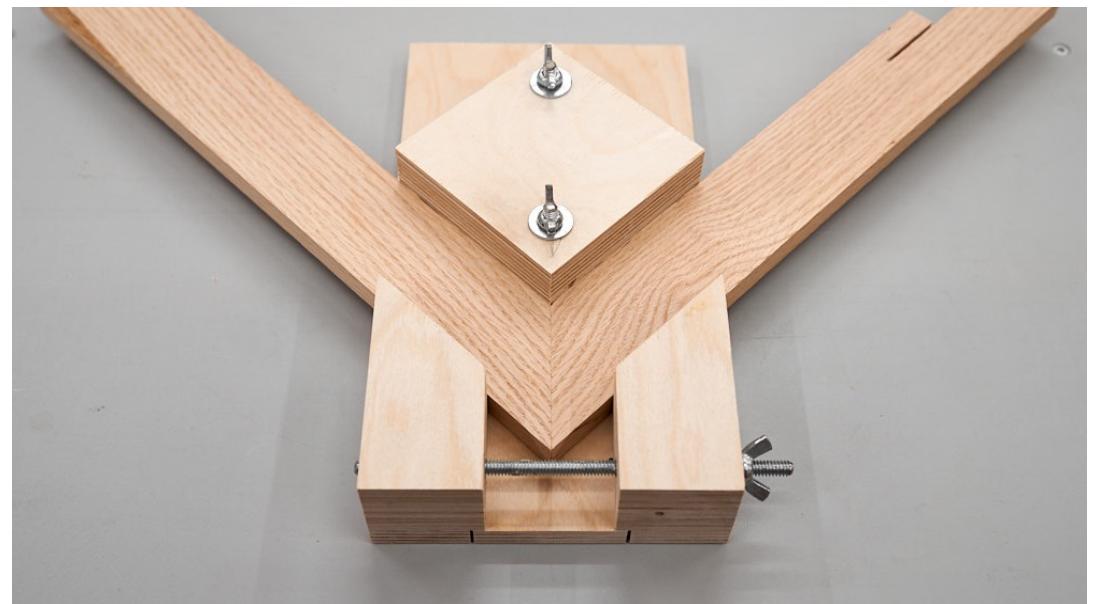
Do one thing well

Decoupling

# Jigs

In woodworking, a template used to get accurate and uniform results

- Consistency
  - Drill holes that are evenly spaced apart
- Safety:
  - Keep fingers away from blades
- Correctness:
  - Keep cabinet corners square
- Convenience:
  - Avoid adjusting guards and fences when cutting



A Jig used to clamp exert force on the corner miter to allow glue to dry.

<https://ibuildit.ca/projects/how-to-make-a-corner-clamp/>

# Software Jigs

- Code Generators
- Kafka -> JQ -> SQLite
- Bash/zsh aliases
- GitHub Repo Configurations

# Example

```
alias mongo_dev_transfer_plan_manager_admin=
"echo \"Logging in as ADMIN in \033[0;32mDEV\033[0m, be careful!\\"; \
mongosh \$(vault read -field=admin_mongo_uri secret/project/dev/transfer_plan_manager)"
```

```
→ mongo_dev_transfer_plan_manager_admin
Logging in as ADMIN in DEV, be careful!
Current Mongosh Log ID: 665f754279cce366b115aff0
Connecting to:      mongodb://<credentials>@[REDACTED]:27017,[REDACTED]:27017,
Using MongoDB:     4.4.14
Using Mongosh:     2.2.6
```

# Removing Ambiguity



**Gregory.Case** 4:29 PM

Widget Maker 3000 is back up and running



**Gregory.Case** 4:07 PM

Now that Kafka is healthy, I re-ran the pipelines for the widget maker 3000 services!



**Gregory.Case** 4:10 PM

Now that Kafka is healthy, I re-ran these pipelines:

```
tapctl run pipeline widgetmaker-keymaster -e prod --lastSuccess --force  
tapctl run pipeline widgetmaker-gatekeeper -e prod --lastSuccess --force
```



# Thinking about Automation

“I restarted the instance”



- Unclear on the reasons
- “Wack-a-mole” Operational Model
- My cat could have done that

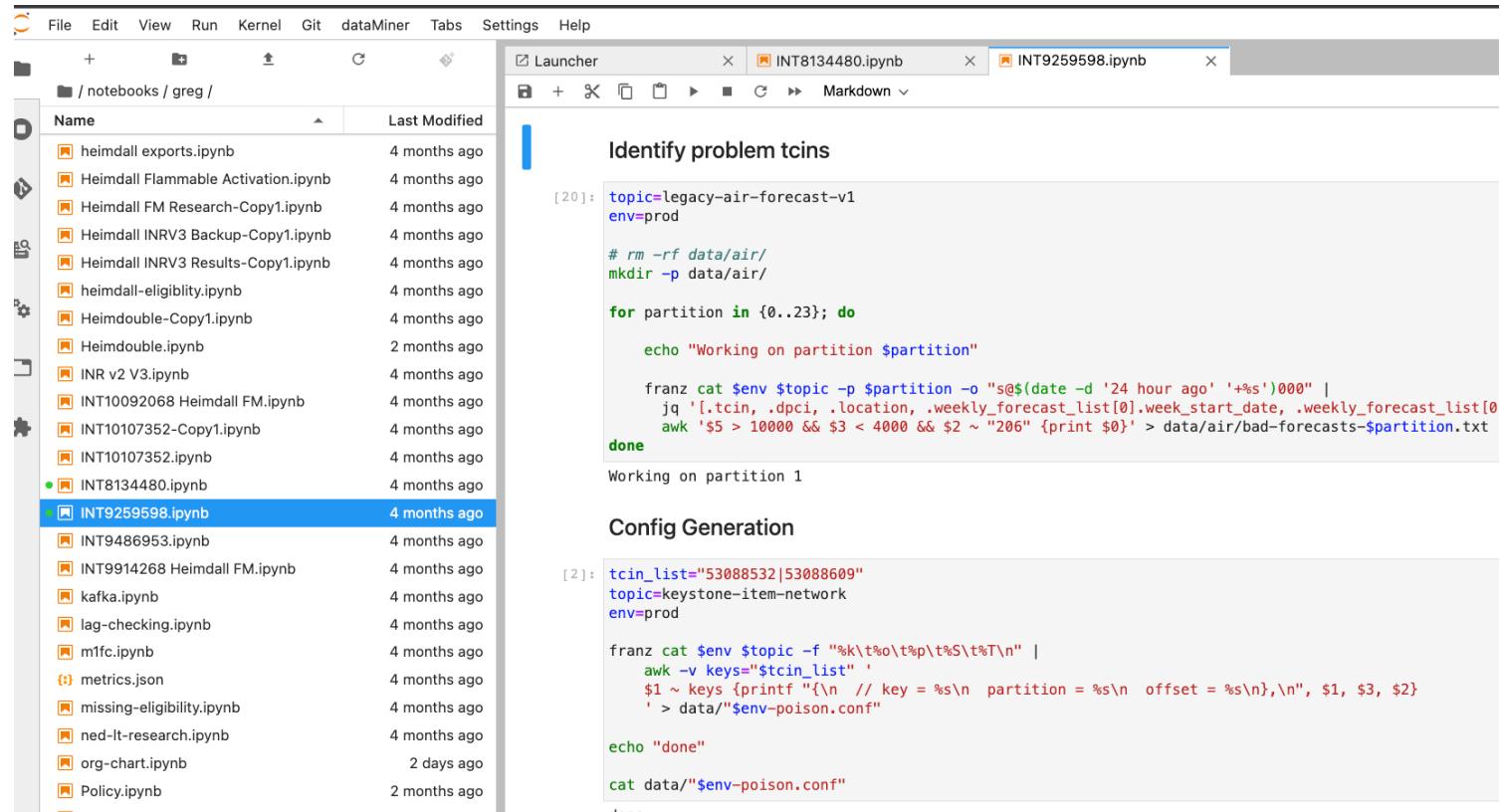
“I wrote a script to detect and terminate unhealthy instances”



- Demonstrates understanding of the problem
- Signals “Promo Ready”
- Frightens other instances into submission

# Tips

Have a cheat sheet  
Use Jupyter Notebooks!  
Track Incidents  
Treat it like real software  
(because it is)



The screenshot shows a Jupyter Notebook interface. On the left is a file browser window titled 'notebooks / greg /' showing a list of Jupyter notebooks. The notebook 'INT9259598.ipynb' is currently selected. On the right are two code cells. The top cell, labeled [20], contains a script to identify problematic partitions in a Kafka topic. The bottom cell, labeled [2], contains a script to generate configuration files for Kafka partitions.

```
[20]: topic=legacy-air-forecast-v1
env=prod

# rm -rf data/air/
mkdir -p data/air/

for partition in {0..23}; do
    echo "Working on partition $partition"

    franz cat $env $topic -p $partition -o "$(@date -d '24 hour ago' '+%s')000" |
        jq '[.tcin, .dpcl, .location, .weekly_forecast_list[0].week_start_date, .weekly_forecast_list[0].forecast] | .tcin = $partition" |
        awk '$5 > 10000 && $3 < 4000 && $2 ~ "206" {print $0}' > data/air/bad-forecasts-$partition.txt
done

Working on partition 1

Config Generation

[2]: tcin_list="53088532|53088609"
topic=keystone-item-network
env=prod

franz cat $env $topic -f "%k\t%o\t%p\t%S\t%T\n" |
    awk -v keys="$tcin_list" '
        $1 ~ keys {printf "%s\n // key = %s\n    partition = %s\n    offset = %s\n", $1, $3, $2, $4}
    ' > data/"$env-poison.conf"

echo "done"

cat data/"$env-poison.conf"
done
```

# Examples

Diagram Generation

API -> SQL “Jig”

Helper Functions

# Your Turn!

45 Minutes at 3¢ per second =

\$81

