

From SAS to Cursor: Vibe Coding into SAS, R & Python

Kevin Lee



Disclaimer

The views and opinions presented here represent those of the speaker and should not be considered to represent any companies or organizations.

Agenda

- Introduction of Vibe-coding
- Why is it so Hyped?
- Current Coding in Biometrics
- Programming workflow using Vibe-Coding
- Vibe coding AI Agents in Industry - Cursor and Windsurf, github copilot
- Customized Vibe coding AI Agent for Biometrics
- Benefits and Risks
- Future of Biometrics in the era of Vibe-coding



Do you use Vibe-coding?



Where “Vibe-Coding” come from?

“fully giv[ing] in to the vibes, embrac[ing] exponentials, and forget[ting] that the code even exists.”

by Andrej Karpathy,
co-founder of OpenAI
on Feb, 2025



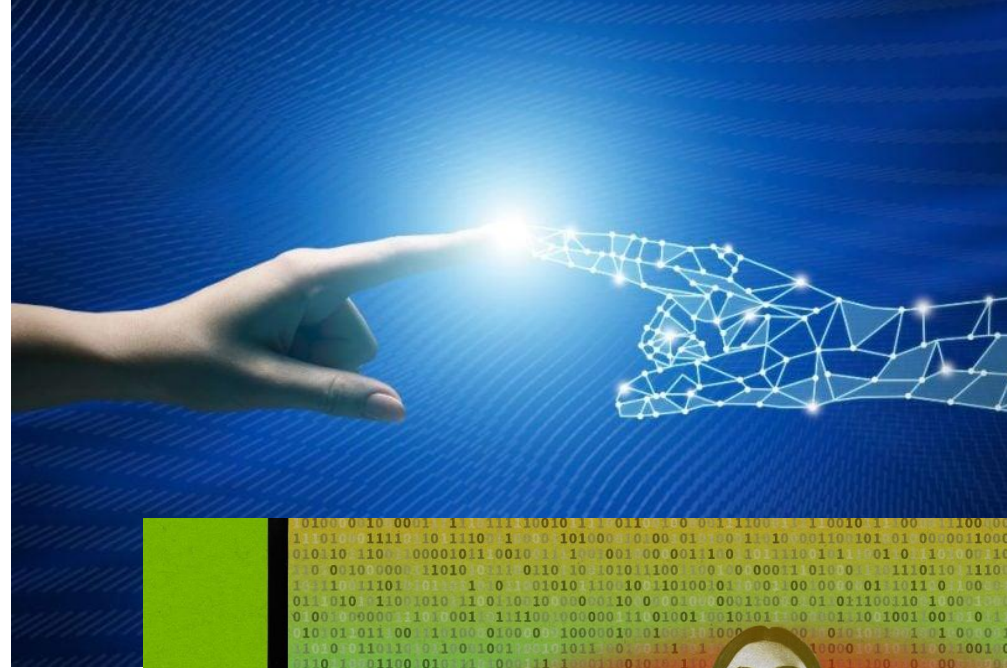
“There's a new kind of coding I call "vibe coding," where you fully give in to the vibes.”

-Andrej Karapathy

EDUCATION **NEXT** 

What is Vibe-Coding?

- Coding with AI
- Vibe : A flow between human logic and AI coding agents
- Programmers do
 - Prompt for the codes
 - Review the codes



The impact of Vibe-Coding?

- Satya Nadella, CEO of Microsoft

“I’d say maybe 20%, 30% of the code that is inside of our repos today and some of our projects are probably all written by software,”

- Sundar Pichai, CEO of Google

"AI generates more than 30% of Google’s code,”

- Mark Zuckerberg, CEO of Meta

“Our bet is sort of that in the next year probably...maybe 50% of the development is going to be done by AI, as opposed to people, and then that will just kind of increase from there,”

Triad of Stat Programming in Biometrics

SAS

- Regulatory Backbone
- Majority on SDTM, ADaM, TFL

R

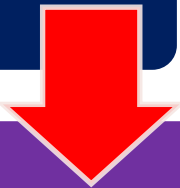
- Data Visualization
- Shiny
- Beginning in SDTM, ADaM, TFL

Python

- Data Pipeline
- AI/ML
- Utilities function

Change in Programming Workflow

Programming Workflow - Traditional



Programming Workflow using Vibe Coding (ChatGPT)



Programming Workflow using Vibe Coding AI Agents

Traditional Programming workflow



ChatGPT impact on programming

- Code Development
- Code Debugging
- Lowering the boundary of languages (e.g., SAS, R and Python) – vibe coding

ChatGPT: Codes Development – R

Prompt 1:

Write a **R script** that groups data by "TRTA" and "AVISIT" columns in "advs.xpt"

and calculates the mean of "AVAL" column for each group at PARAMCD = 'WEIGHT'

Prompt 2:

Write a **R script** that groups data by actual treatment and actual visit in "advs.xpt"

and calculates the mean for each group at weight.

Use ADaM CDISC variable.

Output :

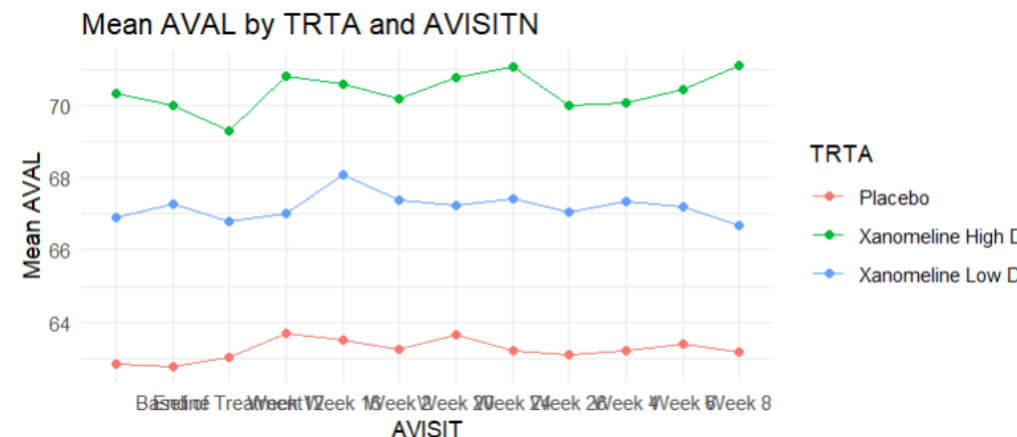
```
library(haven) # For reading XPT files
library(dplyr) # For data manipulation
```

```
# Read the ADaM dataset
advs <- read_xpt("advs.xpt")
```

```
# Process the data
result <- advs %>%
  filter(PARAMCD == "WEIGHT") %>% # Filter for PARAMCD = 'WEIGHT'
  group_by(TRTA, AVISIT) %>%      # Group by TRTA and AVISIT
  summarise(mean_AVAL = mean(AVAL, na.rm = TRUE)) %>%
  # Calculate mean of AVAL
```

```
ungroup()
```

```
# Print the result
print(result)
```



ChatGTP: Codes Development – SAS

Prompt :

Write a **SAS script** that groups data by “TRTA” and “AVISIT” columns in "advs.xpt"

and calculates the mean of “AVAL” column for each group at PARAMCD =‘WEIGHT’

Output :

```
libname mydata xport 'path_to_advs.xpt'; /* Replace with the actual path to advs.xpt */
```

```
data weight_data;  
  set mydata.advs;  
  if PARAMCD = 'WEIGHT'; /* Filter data for PARAMCD = 'WEIGHT' */  
run;
```

```
proc sql;  
  create table weight_means as  
  select TRTA, AVISIT,  
         mean(AVAL) as mean_weight  
  from weight_data  
  group by TRTA, AVISIT;  
quit;
```

```
proc print data=weight_means;  
run;
```

ChatGPT: Codes Development – Python

Prompt :

Write a **Python script** that groups data by “TRTA” and “AVISIT” columns in “adv.s.xpt”

and calculates the mean of “AVAL” column for each group at PARAMCD = ‘WEIGHT’

Output :

```
import pandas as pd
import xport

# Load the .xpt file using xport
with open('adv.s.xpt', 'rb') as f:
    data = xport.load(f)

# Convert the xport data to a pandas DataFrame
df = pd.DataFrame(data['ADVS'])

# Filter the data for PARAMCD = 'WEIGHT'
df_weight = df[df['PARAMCD'] == 'WEIGHT']

# Group by TRTA and AVISIT, then calculate the mean of AVAL
grouped_data = df_weight.groupby(['TRTA', 'AVISIT'])['AVAL'].mean().reset_index()

# Output the grouped data
print(grouped_data)
```

ChatGPT: Codes Debugging – Python

Prompt :

Debug below <python codes> using <error message>.

```
<python codes>:  
import pandas as pd  
import xport
```

```
# Load the .xpt file using xport  
with open('adv.s.xpt', 'rb') as f:  
    data = xport.load(f)
```

```
# Convert the xport data to a pandas DataFrame  
df = pd.DataFrame(data['ADVS'])  
# Filter the data for PARAMCD = 'WEIGHT'  
df_weight = df[df['PARAMCD'] == 'WEIGHT']
```

```
# Group by TRTA and AVISIT, then calculate the  
mean of AVAL
```

```
grouped_data = df_weight.groupby(['TRTA',  
'AVISIT'])['AVAL'].mean().reset_index()
```

```
# Output the grouped data  
print(grouped_data)  
</python codes>
```

```
<error message>  
TypeError: Could not convert ['aval'] to numeric  
</error message>
```

Output:

```
# Convert AVAL to numeric  
df_weight['AVAL'] = pd.to_numeric(df_weight['AVAL'],  
errors='coerce')
```

Programming workflow using Vibe Coding - ChatGPT



Question :

- Can we automate this flow further?

AI Agent

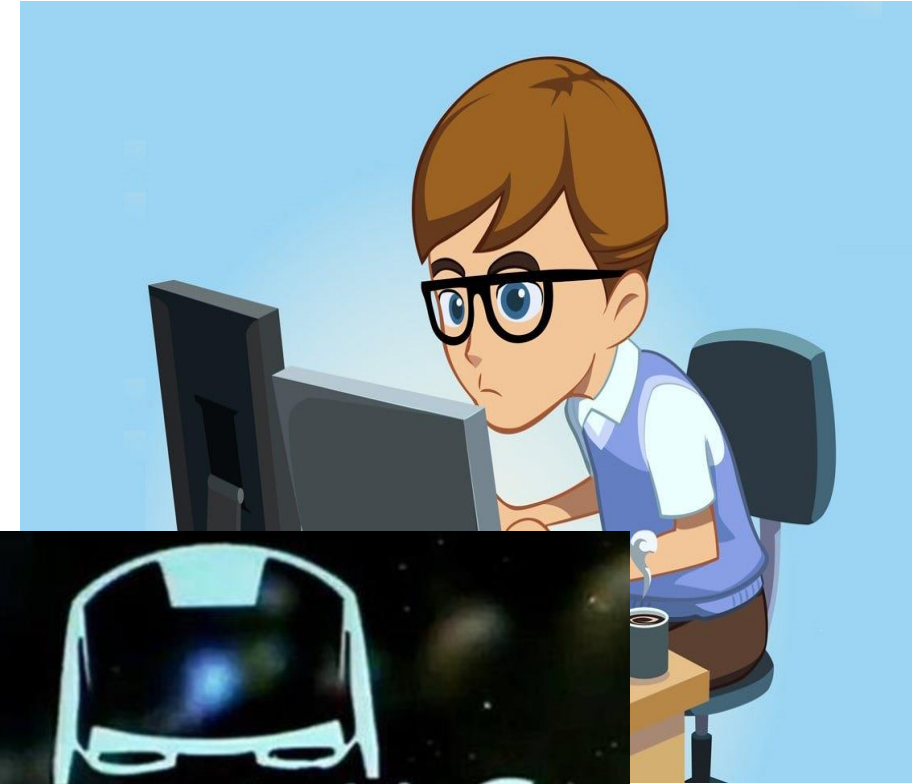
- System that can interact with its environment, collect data, and **autonomously** perform tasks and make decisions to achieve the goals.
- Main Components
 - LLM (e.g., ChatGPT5.2, Claude4.5)
 - Prompts: (e.g., instruction for AI Agent –
**'You are stat programmer.
Use the input and generate SAS output.'**

Follow below steps:

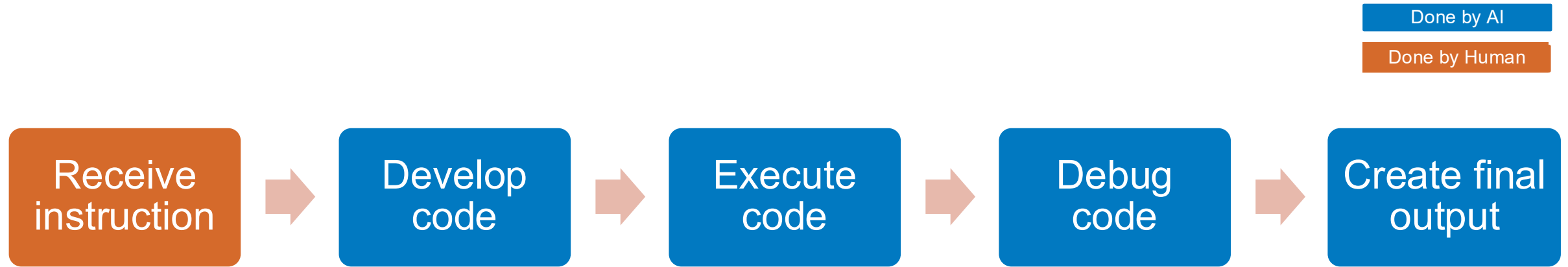
- 1. Generate SAS codes based on input**
- 2. Execute generated SAS codes using SAS Studio.**
- 3. If there are error messages, debug SAS codes.**
- 4. Rerun SAS codes to create the output.**

'Do Step 2 to 4 until there are no error message.'

- Tools (e.g., SAS Studio, R Studio, Python)
- Memory : short-term /long-term memory/ RAG (e.g., previous SAS codes)



Programming workflow using AI Agents



Question :

- What is the problem in this picture / workflow?

ChatGPT or AI Agent is like a very smart, capable, recently graduated intern/new hire.



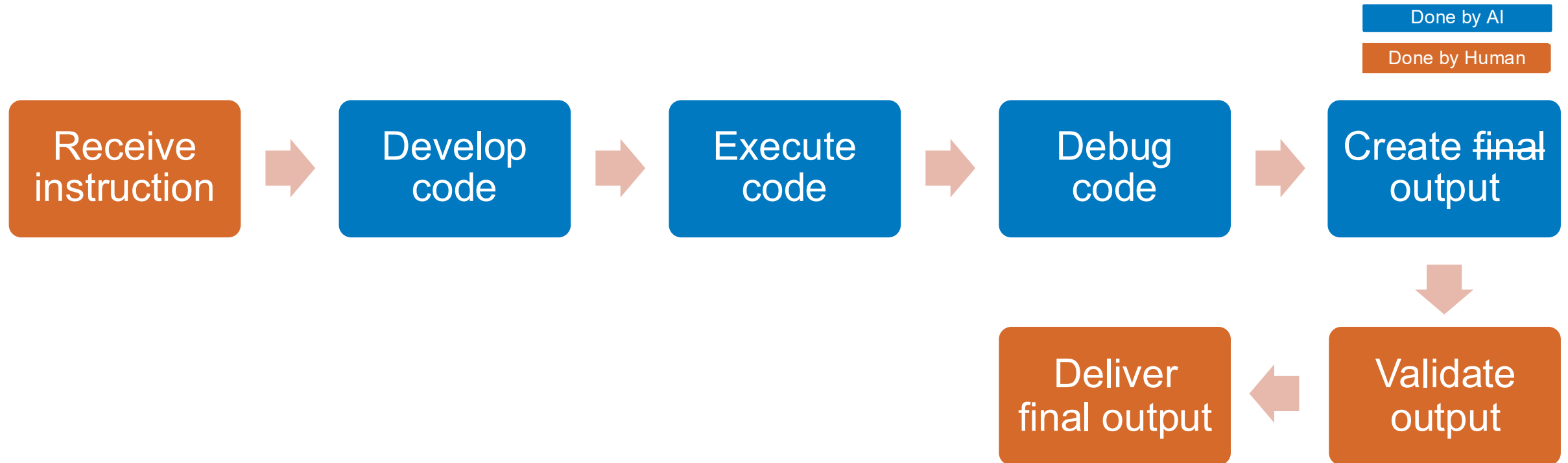
=



What will we do with the results from an intern?



Programming workflow using Vibe Coding AI Agents with human involvement – **AI Agentic workflow**

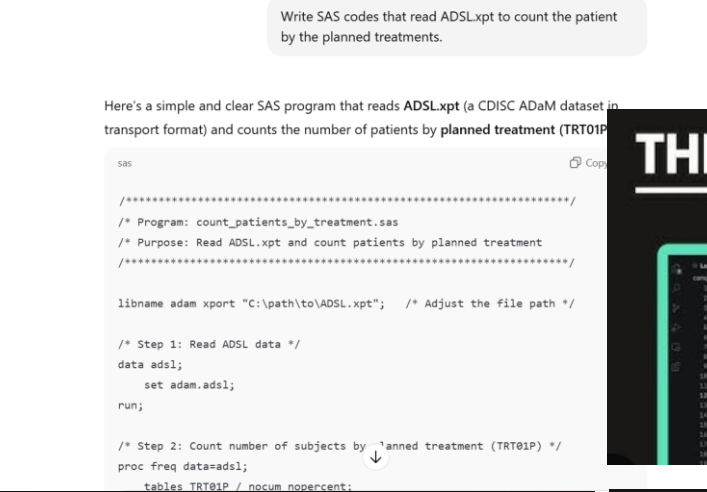
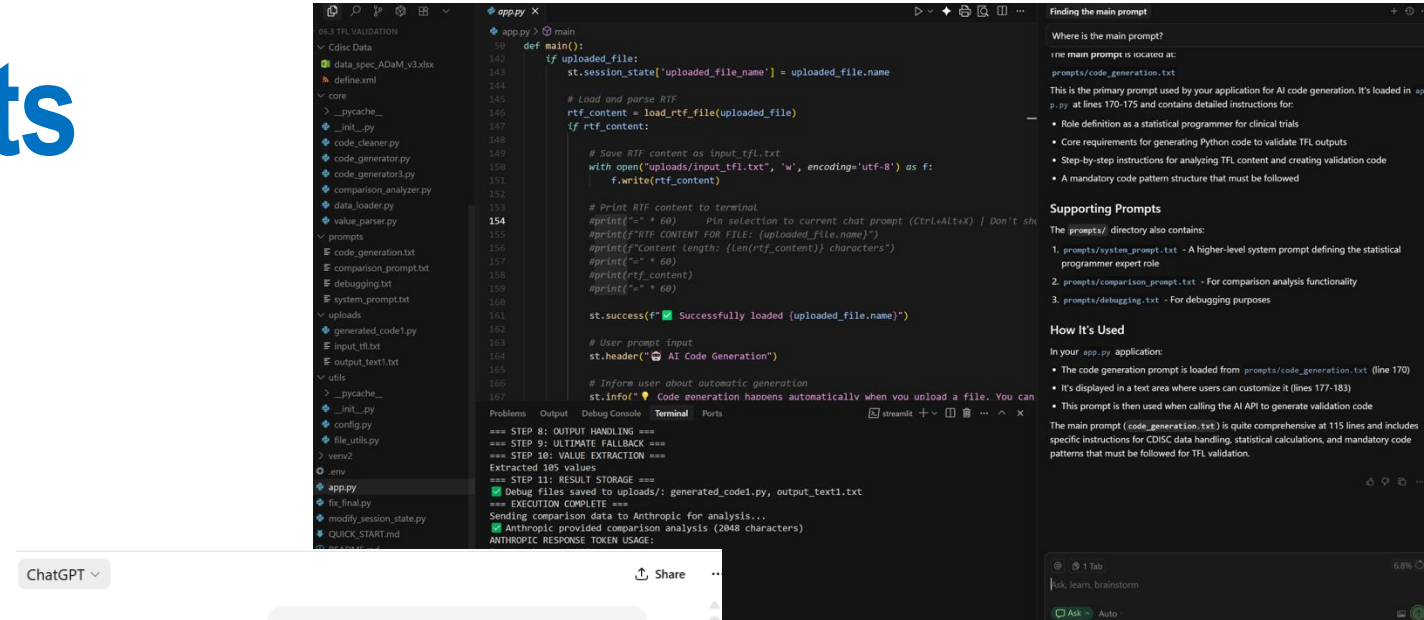


Vibe Coding AI Agents Tools

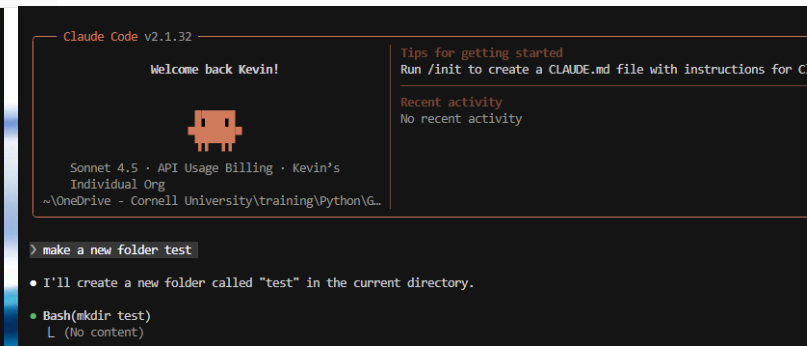
- **ChatGPT / Claude / Gemini** – web-based prompt engines for ideation
- **Cursor** – AI-native IDE built around conversational programming
- **Windsurf** – context-aware code editing and real-time collaboration
- **GitHub Copilot** – inline suggestions in VS Code, RStudio, Jupyter
- **Claude Code** – AI coding assistant that works in the terminal to read, edit, and help fix code using natural language



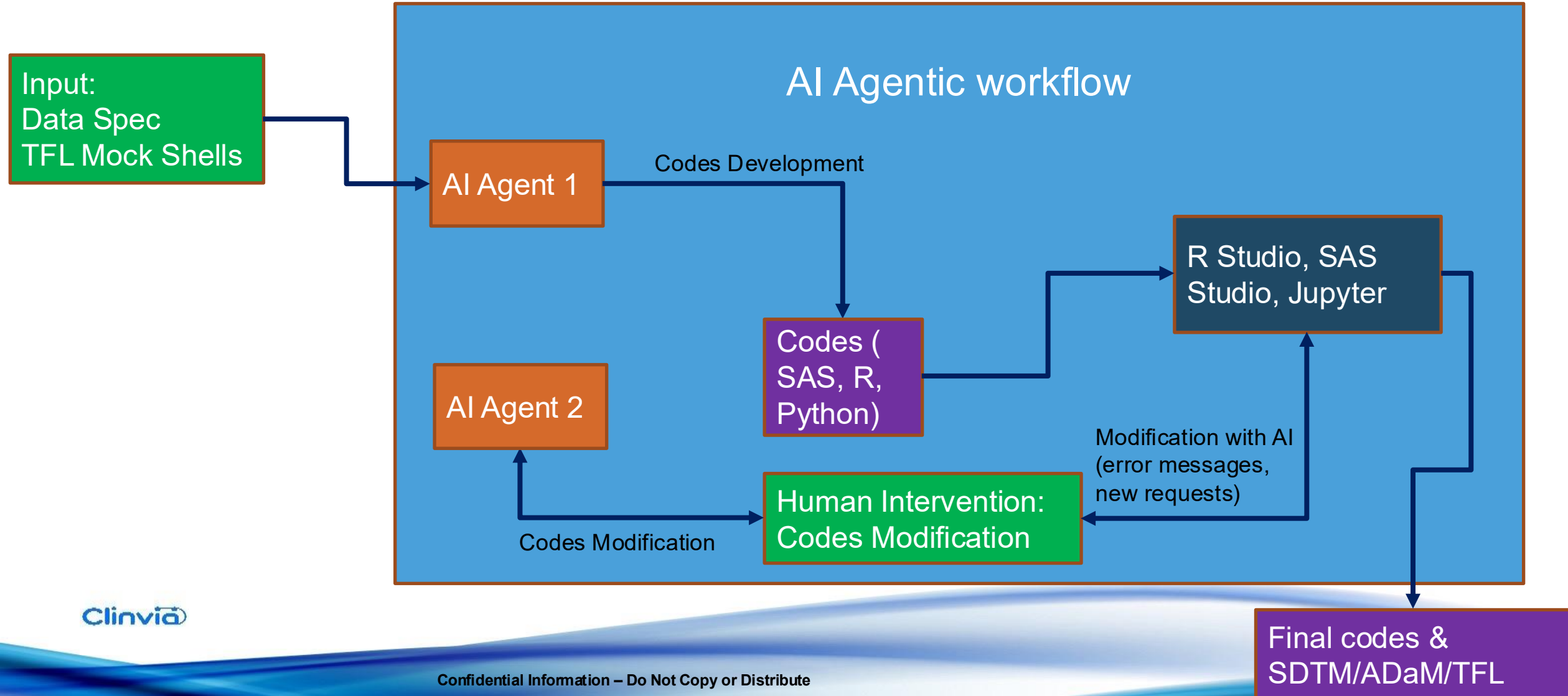
Confidential Information – Do Not



THIS IDE BUILT THE ENTIRE APP



Customized AI Coding Agent



Why customized Vibe Coding Agent for Biometrics works?

- Generic Coding AI Agents lack domain knowledge
- Biometrics = highly specialized: CDISC, ADaM, TLFs, Compliance
- Customized AI Agents embedded with clinical trial logic + GxP validation rules
- Data Privacy and Security
- Dirty Data

Key Benefits on Vibe-coding

- Dramatic reduction in repetitive coding tasks
- Accelerates QC and development cycles (25 to 40%)
- Enables multi-language transition (SAS/R/Python) and consistency
- Democratizes programming across experience levels
- Improves documentation and traceability

Potential Risks on Vibe-Coding

- Hallucinated or Inaccurate code outputs
- Limited domain understanding
- Regulatory & Compliance challenges (Audit trail, GxP-Compliance, Validation, Data Privacy)
- Risk of over-reliance on AI judgment

Governance & Validation framework

- **Regulatory Importance:** Vibe coding outputs must meet the **same quality standards** as traditionally produced results – traceability, reproducibility, accuracy, compliance.
- **Biometrics Leadership Role:** Must establish rigorous, **risk-based validation process/systems** tailored for Vibe-coding agentic workflow & process.
- **Framework Components:**
 - Risk-based validation planning
 - Verification of Vibe-coding workflow output accuracy
 - Application of GxP / GAMP 5 principles
 - Documented human approval and test evidence(e.g., plans, protocols, test scripts/results)
 - Version control and audit trails
 - Package validation



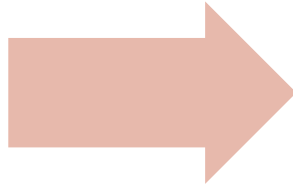
Future of Vibe-Coding and US

- **Change of Coding**
- **Evolving Roles**
- **New Skill Sets Building**
- **AI and Human Collaboration**
- **Leading into the future**

The Change of Coding

Traditional Coding

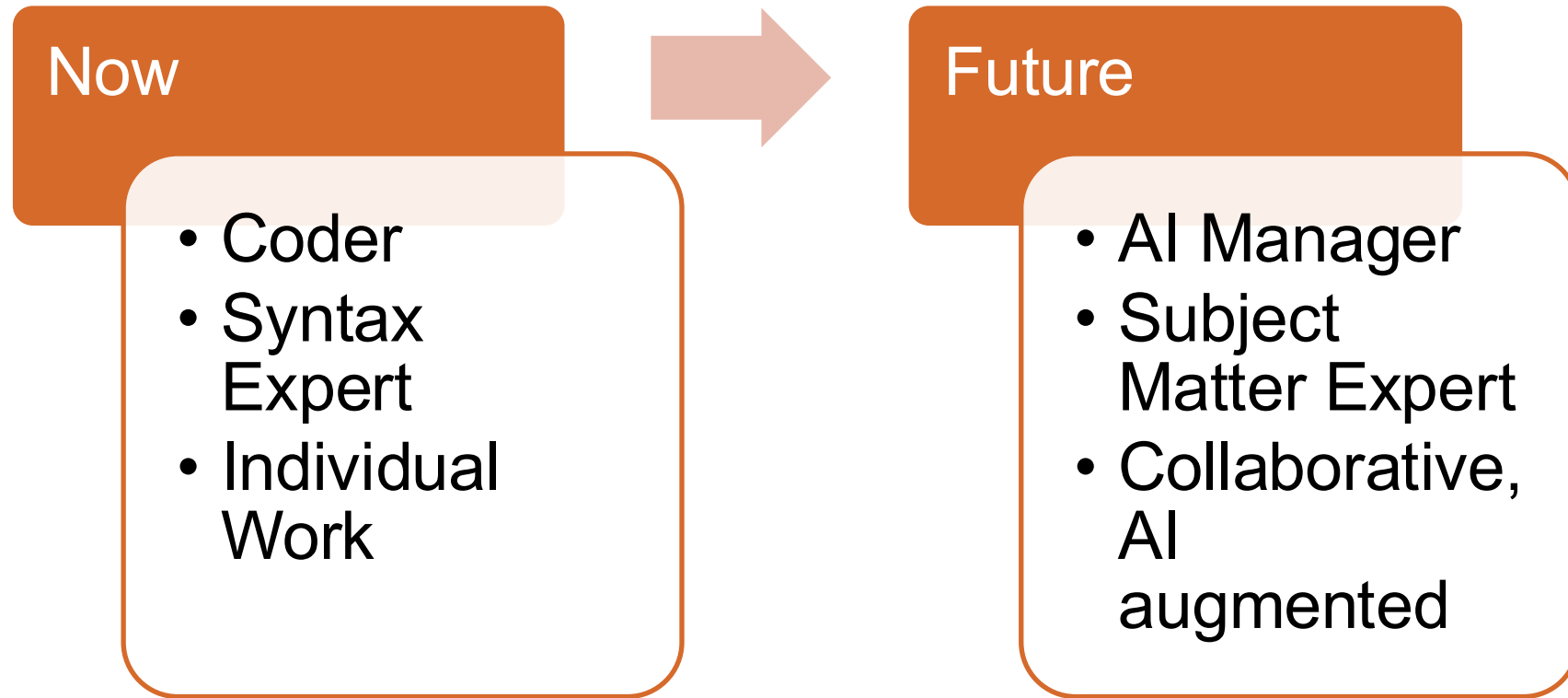
- Manual / Hands-on
- Codes/Rules-based
- Writing codes
- Syntax-driven



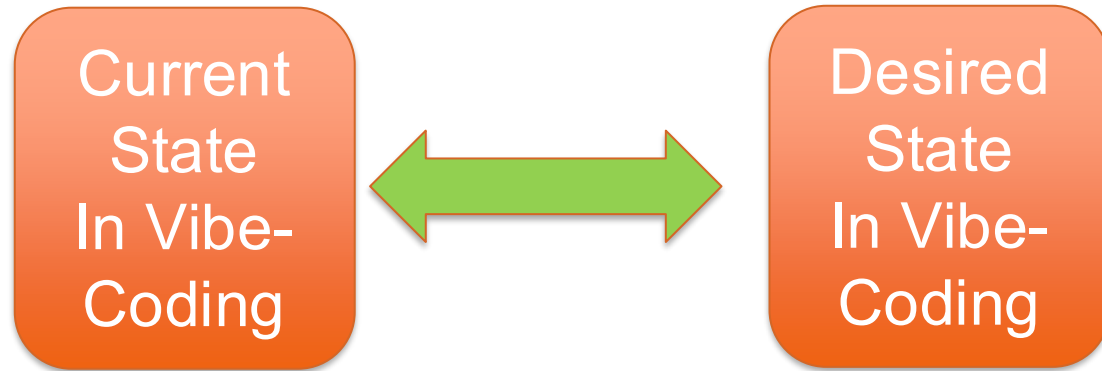
Vibe Coding

- Conversational
- AI-assisted
- Designing workflow
- Context-aware

Evolving Roles for US



New Skill Sets Building



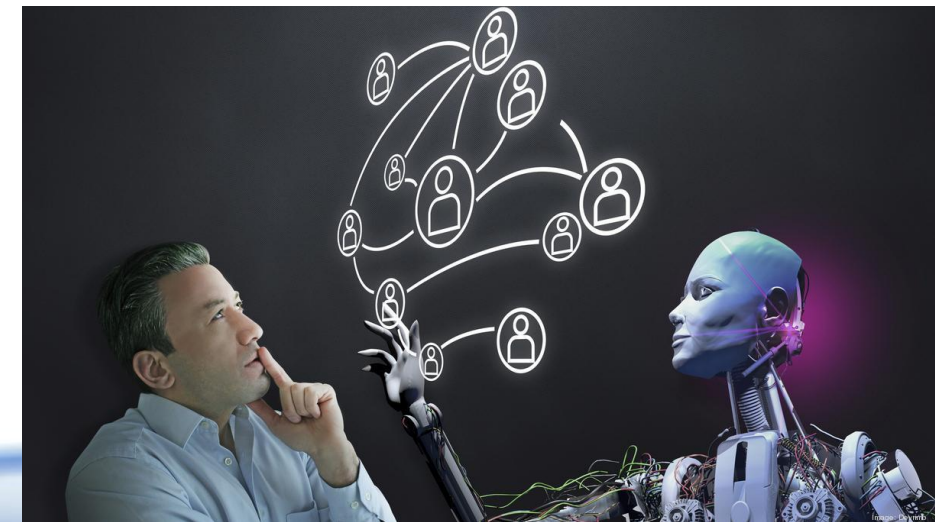
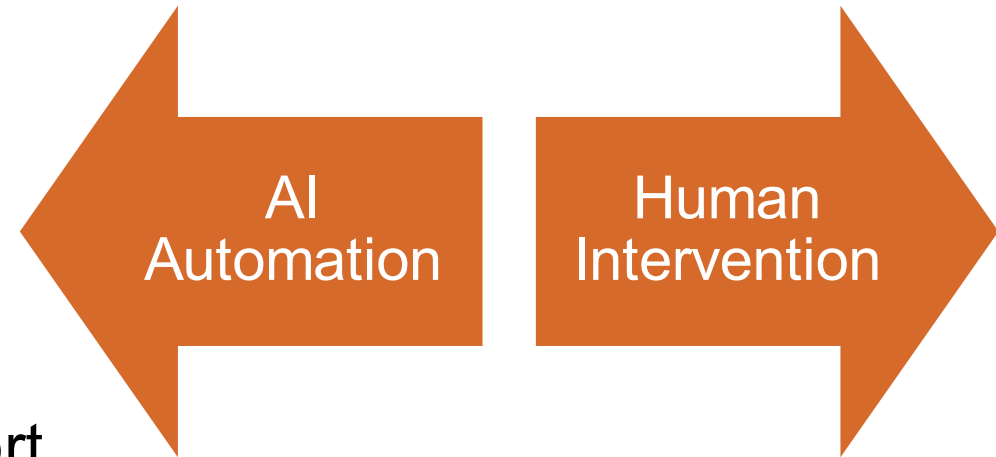
Assessment

- Existing skill gaps
- Implement targeted training initiatives.

- Focus areas for capability building:
 - Prompt engineering
 - Technical proficiency with Vibe Coding tools (e.g., ChatGPT, Copilot, Cursor, Windsulf, Github copilot)
 - Validation skills for Vibe-coding-generated outputs
 - Understanding of Agentic workflow – AI and Human Collaboration Policy
 - More Humane Characteristics – EQ, Creativity, Critical Thinking, Collaboration, Communication

Human vs Vibe Coding Agents Collaboration Policy

- Clear delineation
 - AI-driven activities (Vibe Coding)
 - Human-driven activities
- AI roles
 - **Assistant**: e.g., generating initial code or report templates
 - **Autonomous Operator**: e.g., handling routine coding development and checks.
- Human intervention- required to ensure **oversight** and **accountability**.



Leading into the future – Vibe Coding AI Strategy

- Define Vibe-coding augmented Biometrics Roadmap
- **Position Biometrics as the innovation hub** within clinical development: the bridge between data, analytics, application and AI.
- Shift from “**Submission deliverables**” to “**building intelligent AI powered team.**”
- Lower language barriers : from **SAS to multi-lingual programming.**
Use the best tools for the right job.

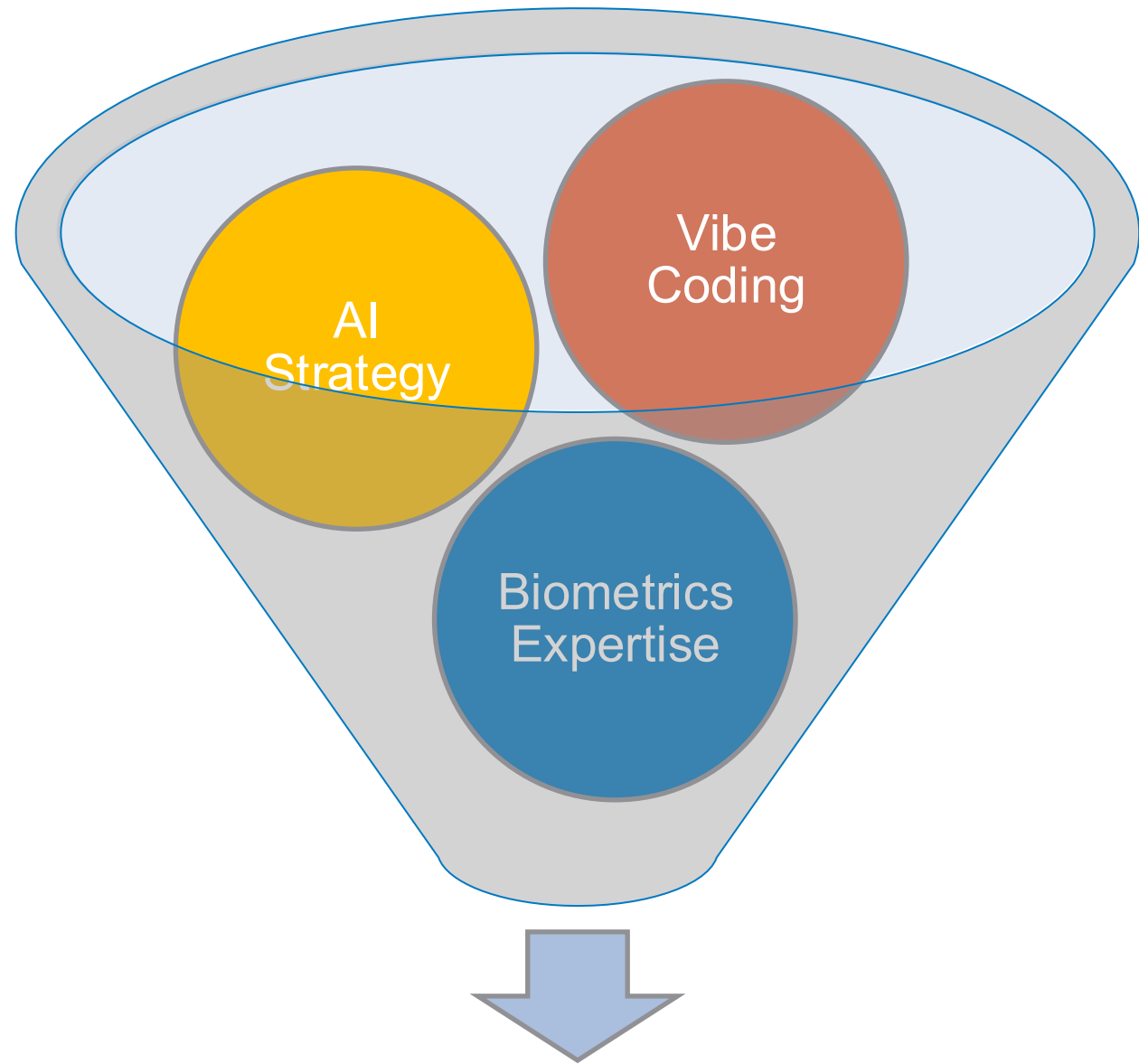
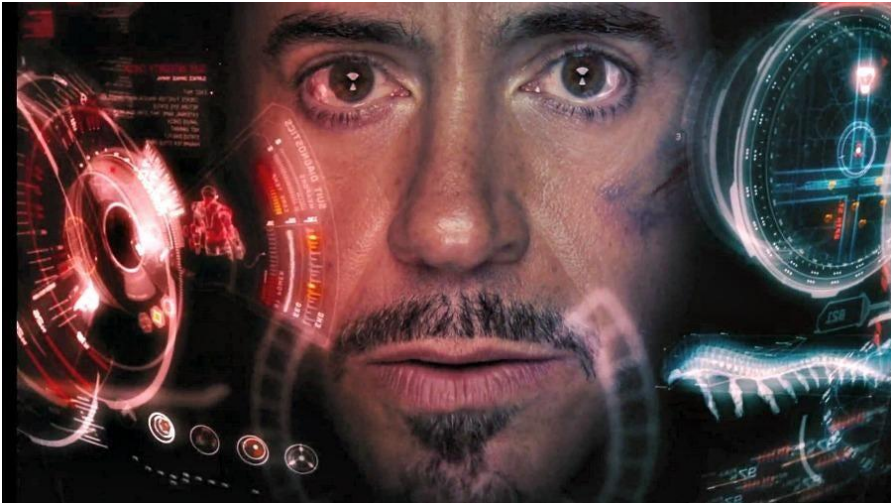
The impact of Vibe-Coding?

- Bain & Company : Average developer time savings: **10–15%** across all engineering tasks.
- Github & Stack Overflow : Efficiency boosts for targeted tasks (boilerplate, documentation, testing), **30–60% time savings**.
- Atlassian found that **99%** of developers reported time savings using AI tools, with **68%** estimating more than 10 hours saved per week.
- JPMorgan Chase measured up to **20% overall software engineering** efficiency gain with AI.
- Federal Reserve Bank of St. Louis : Higher reported savings for daily AI users, who save up to **33% more time per coding hour** by automating repetitive work.



Discussion :

**Will Vibe-coding
replace US or
empower US as a
programmer?**



Vibe-coding powered “Super Biometrics Team (Avenger)”

ANY QUESTIONS



Kevin Lee at kevin@clivia.com

LinkedIn : <https://www.linkedin.com/in/hellokevinlee>



Appendix



Proprietary and Confidential Information – Do Not Copy or Distribute

Programming workflow using Vibe Coding AI Agents with human involvement – **AI Agentic workflow**

