# UX Research Portfolio

Grayson Butcher, PhD

UX Researcher | Behavior Scientist

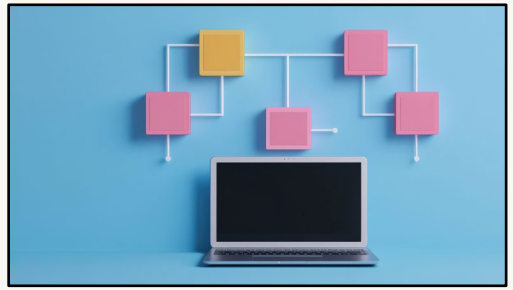
# Table of Contents

**Case Study 1:**

Mentoring Program

**Case Study 2:**

University Training



**Case Study 3:**

Behavioral Scoring App

# Case Study 1:

# Designing a Scalable Mentorship Program

How I transformed an informal student-support effort into a structured, data-driven mentoring program that scaled from 5 to 20+ students and secured ongoing university funding.

# Problem to Solve

Neurodivergent college students were working hard but some still struggled with their academic and personal goals. Few received support, and those who did were struggling with the existing, informal mentoring program. I was tasked with building a better way to support these students.

**The research I conducted had the following goals**:

1. Understand current student experiences, needs, and contextual constraints
2. Identify the conditions and feedback loops that predict meaningful outcomes
3. Define measurable outcome and process metrics, and prototype an improvable, data-informed structure for mentoring.

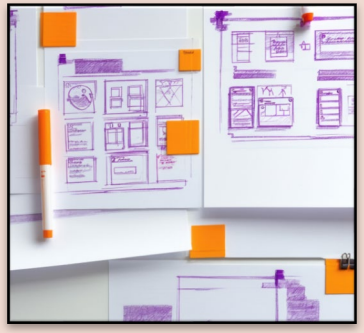
# Approach to Solution

Stakeholder Interviews + Benchmarking

Field Studies

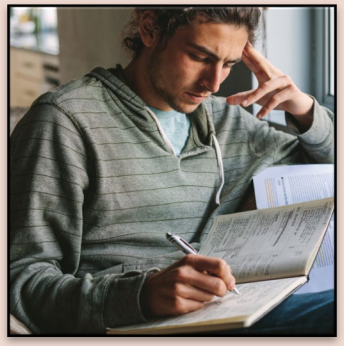
Task Analyses + Systems Mapping

Discovery



Prototype Testing

Design



Diary Studies

Longitudinal Evaluations

Delivery

# Insights

1. **Goal ownership drives engagement.** Explicit, student-led goal-setting produced > 90% attendance and goal progress.

2. **Structure enables autonomy.** Behavior-based logs, visual progress tools, and goals-based feedback improved student goal adherence from 50 to 80%

3. **Scaffolding shapes outcomes.** Templates and metrics reduced off topic drift by > 10 minutes

# Outcomes

1. Program scaled from 5 to 20+ mentees

2. GPA rose > 0.3 points for all students

3. Official workplace inclusion funding contract secured + expanded

4. Students reported high satisfaction, greater confidence managing goals, and meaningful progress on connection/community goals.

# Case Study 2:

# Neurodiversity Training

# Case Study 2:

# Designing a University-Wide Training Program

How I used UXR to design and launch a research-informed, university-wide training program on neurodiversity for faculty, staff, and students.

# Problem to Solve

Faculty and staff wanted to support students but often misunderstood what “neurodiversity” meant in practice. Existing resources were inconsistent and compliance-focused, leaving learners confused, hesitant, or afraid of “saying the wrong thing.” Students reported that even well-intentioned interactions felt stigmatizing or inflexible.

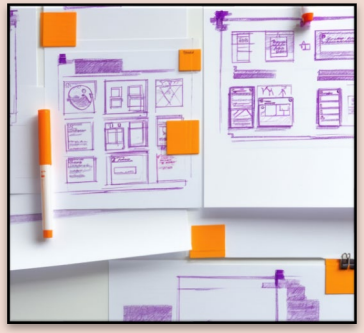
**The research I conducted had the following goals**:

1. Understand how stakeholders current define, discuss, and act on neurodiversity
2. Identify misconceptions, sensitivities, and barriers to inclusive behavior
3. Design an engaging and measurable learning experience that translates theory into daily action

# Approach to Solution

Stakeholder Interviews + Landscape Review

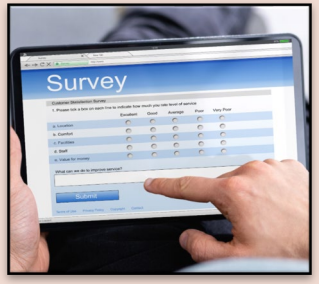
Task Analyses



Prototyping

Usability testing

Mixed-methods evaluations




Post-training surveys

Discovery → Design → Delivery

# Insights

1. **Clarity accelerates comprehension.**
   Participants learned fastest when sessions began with motivational and theoretical anchors, followed by concrete, scenario-based practice.
2. **Interactivity & Retention**
   Activities that required learners to actively respond to real-world vignettes produced excellent transfer (> 85% accuracy)
3. **Feedback Loop**
   Post-session reflections revealed lingering uncertainty that informed refined examples and a quick-reference glossary.

# Outcomes

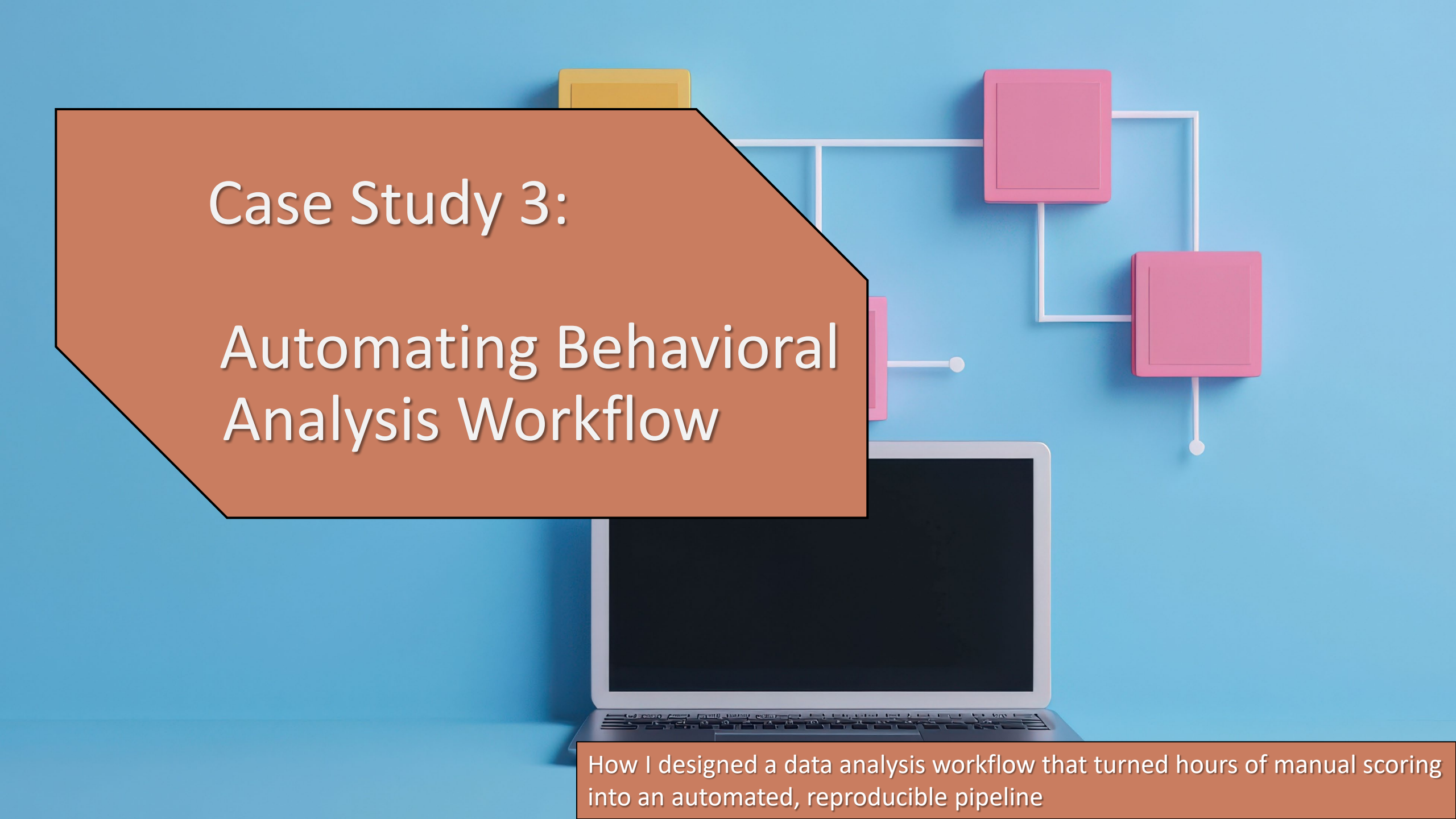
1. Training deployed university-wide (live + online) and adopted by the Inclusion Office as a standing professional-development module.
2. Post-training surveys showed > 30% gains in confidence using inclusive language and recognizing neurodivergent strengths.
3. Material integrated into onboarding for faculty and staff, creating shared language and consistent practices across departments.

# Case Study 3:

# Building a Local App for Research Lab

# Case Study 3:

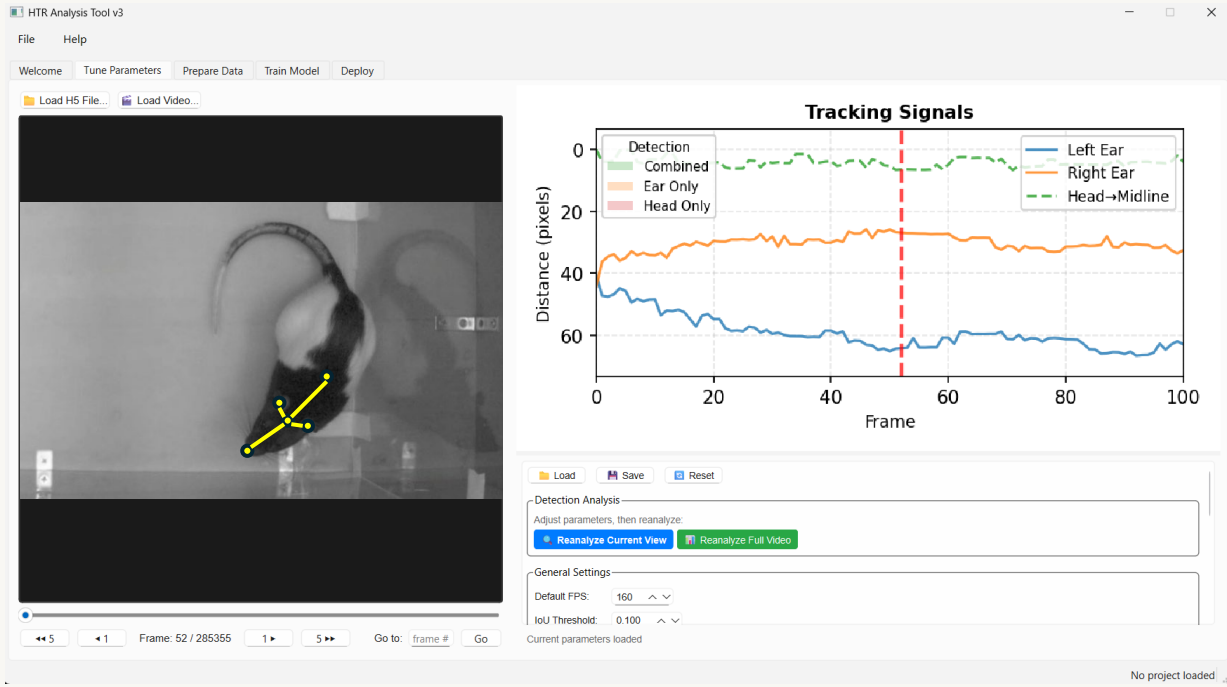
# Automating Behavioral Analysis Workflow



How I designed a data analysis workflow that turned hours of manual scoring into an automated, reproducible pipeline

# Problem to Solve

Researchers relied on manual video scoring, an error-prone process taking hours per video. We needed a scalable, transparent system that could automatically and accurately detect subtle, infrequent behavior.




## The research I conducted had the following goals:

- Understand how early users interacted with the multi-script version to uncover friction points and ideal workflows.

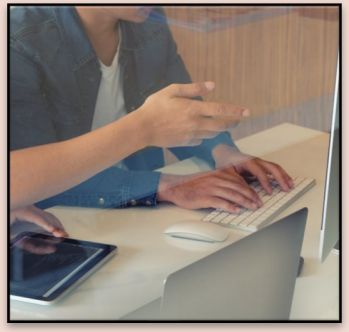
2. Identify and design interface features that make automated tools feel interpretable and credible to non-coding users.

3. Use observed pain points and feedback to shape the interface structure, control layout, and visual feedback of the GUI version.

# Approach to Solution

Contextual Inquiry

Task Analysis



Usability testing


A/B testing

Qualitative Interviews

Discovery → Design → Delivery

# Insights

1. **Streamlined onboarding improves adoption.** Early GUI tests revealed hesitation during first runs. Reorganizing tabs and adding a visual workflow guide cut setup and execution time by > 50 %.
2. **Clear parameters drive consistency.** Users often guessed settings. Adding in-app tooltips and concise variable definitions reduced setup errors and standardized training across users.
3. **Real-time feedback enhances flow.** Progress bars and live status indicators minimized idle uncertainty, making the tool feel faster and more predictable.
4. **Iterative design enabled reliability.** After three prototype cycles, average setup and run times dropped **50–75 %**, and non-coding lab members could complete full analyses without supervision.

# Outcomes

1. The GUI version reduced analysis time from several hours per video to **under 30 minutes** per batch process of 8+ videos.
2. Non-coding lab members could now train and run models independently, expanding automation access across the team.
3. The project is being adopted across experiments as a standardized analysis tool, improving reproducibility and efficiency.
4. The open-source release on GitHub allows others to replicate or extend the approach.