GSDC
Global Skill Development Council

REAL-WORLD CASE STUDIES IN AI TESTING

www.gsdcouncil.org

CASE STUDY 1: Amazon's Recruiting AI — When Bias Goes Undetected

Background

Amazon built an AI-powered recruiting tool starting in 2014 to automate the screening of job applications. It was trained on 10 years of historical resumes submitted to Amazon — a company where, at the time, the vast majority of technical hires were male. The system was designed to score candidates on a 1-to-5 star scale and surface the best applicants automatically.

What Went Wrong

The model learned that male candidates had historically been preferred — because the training data reflected a historically male-dominated hiring pattern.

It penalized resumes containing the word "women's" — as in "women's chess club" or "women's college." It also downgraded graduates of all-women's colleges.

The bias was **systematic and invisible** — the model never explicitly considered gender, but learned gender as a proxy through other features.

Most critically: the bias was not discovered through testing before deployment — it was discovered after the tool was in use.

The Testing Failures

- No formal fairness testing was conducted before deployment
- There were no demographic breakdowns of model performance across candidate groups
- No counterfactual testing was done — no one tested: "What happens if I change gender-associated words?"
- Monitoring for disparate outcomes by demographic group was not in place
- The bias was only discovered when humans reviewed the tool's recommendations and noticed the pattern

Amazon Recruiting AI — Result & Lessons

Amazon shut down the tool in 2018. The reputational damage was significant. The case became one of the most cited examples globally of algorithmic bias in high-stakes decision-making.

What Good AI Testing Would Have Caught

Bias Testing

Comparing model scores across male and female candidates with equivalent qualifications

Counterfactual Tests

Showing score changes when gender-coded language was altered

Fairness Metrics

Showing disparate impact on female applicants

Human Review

Review of top recommendations before deployment, structured to detect demographic patterns

Key Takeaways

- Training data that reflects historical bias produces models that perpetuate and automate that bias
- Bias in AI is rarely explicit — it is embedded in features that correlate with protected attributes
- Fairness testing must happen **before deployment**, not after discovery in production
- Human review of AI recommendations is not optional in high-stakes decisions
- The cost of not testing for bias: legal exposure, reputational damage, and real harm to real people

CASE STUDY 2: Healthcare AI Misses Sicker Black Patients — Bias in Medical Algorithms

Background

A 2019 study published in *Science* analyzed a widely used commercial healthcare algorithm deployed across US hospitals to identify patients who would benefit from additional care management. The algorithm was used to manage the care of approximately 200 million people in the United States.

What Went Wrong

- The algorithm used **healthcare spending** as a proxy for **healthcare need** — assuming that patients who cost more to treat must be sicker
- In reality, Black patients faced systemic barriers to accessing care — they spent less on healthcare not because they were healthier, but because they had less access
- The result: the algorithm systematically assigned **lower risk scores** to Black patients than to equally sick White patients
- Black patients needed to be considerably sicker than White patients to receive the same risk score and qualify for care management programs

The Testing Gap

The algorithm's developers tested performance overall — accuracy metrics looked acceptable at the aggregate level

No disaggregated performance analysis was conducted by race

There was no test of whether the proxy variable (spending) was an equitable stand-in for the target variable (health need) across demographic groups

The assumption that "spending = need" was never challenged through fairness testing

Healthcare AI — Result & Lessons

Researchers estimated the bias reduced the number of Black patients identified for additional care by approximately 50%. The algorithm vendor subsequently updated the tool after the study's publication.

What Good AI Testing Would Have Caught

Disaggregated Analysis

Performance analysis by race showing different error rates across demographic groups

Proxy Variable Testing

Is spending an equitable proxy for need across groups?

False Negative Rate

Comparison of false negative rates across demographic groups

Clinical Expert Review

Review of cases where the model's recommendation diverged from clinical judgment

Key Takeaways

- Aggregate accuracy metrics can hide severe bias — always disaggregate performance by demographic group
- Proxy variables are a major source of hidden bias — test whether your proxy is equitably valid across groups
- Healthcare AI failures have direct life-or-death consequences — the stakes of inadequate fairness testing are not theoretical
- Algorithmic bias in healthcare is not neutral — it compounds existing health inequities
- Fairness is not a feature to add later — it must be tested from the start

CASE STUDY 3: Uber's Surge Pricing Model — Data Drift in the Wild

Background

Uber's dynamic pricing algorithm (surge pricing) uses real-time demand and supply signals to set ride prices. The model is trained on historical patterns of demand — when, where, and how demand spikes.

What Went Wrong

During the 2017 New York City terrorist attack in Manhattan, Uber's algorithm detected a massive spike in demand in the affected area — and triggered surge pricing. Prices multiplied dramatically in a zone where people were fleeing danger. The algorithm had no mechanism to detect or handle an emergency context.

In a different context, during COVID-19 lockdowns, Uber's demand models trained on years of pre-pandemic patterns became immediately and profoundly obsolete — a textbook case of sudden concept drift.

The Testing Gap

- The model had no context-aware testing — it could not be tested for behavior in emergency or crisis scenarios because those scenarios were not represented in training data
- There were no edge case tests for extreme, low-probability demand spikes in specific geographic clusters
- No monitoring mechanism existed to detect when incoming data was structurally different from anything the model had been trained on
- Drift monitoring was not designed to trigger on sudden, unprecedented concept shifts

Uber Surge Pricing — Result & Lessons

The surge pricing during the 2017 attack caused significant public and media backlash. Uber later committed to disabling surge pricing during public emergencies. The COVID-19 demand collapse required emergency model retraining across Uber's entire operations.

What Good AI Testing Would Have Caught

Scenario-Based Edge Cases

Testing including low-probability, high-impact events

Out-of-Distribution Testing

What does the model do when inputs look nothing like training data?

Clustering Tests

Geographic and temporal clustering tests for anomalous demand patterns

Drift Detection

Automated concept drift detection with human escalation for unusual pattern shifts

Business Rule Overrides

Testing whether safety rules supersede model outputs in crisis contexts

Key Takeaways

- AI models trained on historical patterns will eventually encounter situations those patterns do not cover — test for it
- Sudden concept drift (COVID-19, emergency events) requires both monitoring and human override capability
- Edge case testing should specifically target low-probability, high-impact scenarios
- Business logic and human override systems are part of AI system testing — not separate
- Drift monitoring must be sensitive enough to catch both gradual and sudden distribution changes

CASE STUDY 4: Microsoft's Tay Chatbot — Security & Adversarial Failure

Background

In March 2016, Microsoft launched Tay — an AI chatbot on Twitter designed to learn from conversations with users and mimic the language patterns of a 19-year-old American. Tay was live for approximately 16 hours before Microsoft had to shut it down.

What Went Wrong

- Within hours of launch, coordinated groups of Twitter users discovered that Tay could be manipulated through repeated inputs to produce harmful, offensive, and hateful content
- Users discovered that Tay would repeat and amplify language fed to it — a form of **prompt injection and model manipulation**
- The model had no robust filtering, no adversarial input detection, and no rate-limiting on influence by any single source
- Within 24 hours, Tay was producing racist, sexist, and inflammatory content at scale

The Testing Gap

No **adversarial testing** was conducted — no one tested what happened when users deliberately tried to manipulate the model

No **red team testing** — no internal team attempted to break the system before public launch

No **content safety testing** — there was no systematic test of whether the model could be induced to produce harmful outputs

No **input volume or source weighting** — no testing of whether coordinated input from a small group could dominate the model's learning

No escalation or shutdown mechanism was tested in advance

Microsoft Tay — Result & Lessons

Microsoft shut Tay down within 16 hours, issued a public apology, and the incident became one of the most studied examples of AI safety failure in the industry.

What Good AI Testing Would Have Caught

Red Team Testing

Dedicated team attempting to break the model before launch

Adversarial Input Testing

Systematic attempts to manipulate outputs through crafted inputs

Hate Speech Filters

Harmful content filter testing before any public deployment

Coordinated Input Simulation

What happens if many users send similar manipulative inputs?

Output Monitoring

Automated detection of harmful content categories

Shutdown Procedures

Shutdown and escalation procedures tested end-to-end

Key Takeaways

- Any AI system deployed publicly will be adversarially tested by users — do it yourself first
- Red team testing is not optional for public-facing AI systems
- Content safety is a testing discipline — not a post-launch moderation problem
- Adversarial robustness must be tested before deployment, not discovered in production
- The speed of AI failure in public can be catastrophic — 16 hours from launch to shutdown

CASE STUDY 5: Apple Face ID — When Diversity in Test Data Saved the Product

Background

When Apple developed Face ID for the iPhone X (2017), it invested heavily in building a diverse and representative training dataset for its facial recognition model — specifically to avoid the well-documented pattern of facial recognition systems performing significantly worse on women and people with darker skin tones.

What They Did Right

Diverse Dataset

Apple gathered a dataset of over one billion images from diverse global populations — ensuring representation across age, gender, skin tone, and ethnicity

Disaggregated Testing

They specifically tested performance disaggregated by demographic group before launch — not just overall accuracy

Real-World Variation

They tested performance across real-world variation: glasses, hats, scarves, makeup, facial hair, and different lighting conditions

Explicit Thresholds

They set explicit performance thresholds that had to be met across all demographic groups, not just aggregate performance

Twin Studies

They conducted twin studies specifically to test whether the model could distinguish between identical twins — one of the hardest fairness test cases

Apple Face ID — Result & Key Takeaways

1/1M

False Acceptance Rate

Face ID launched with a false acceptance rate of approximately 1 in 1,000,000 — significantly better than Touch ID's 1 in 50,000

2017

Launch Year

The system has maintained strong performance and trust since its 2017 launch

Performance was substantially more equitable across demographic groups than earlier facial recognition systems.

Key Takeaways

- Representative training data is the most powerful fairness intervention available — diverse data before diverse testing
- Disaggregated performance testing by demographic group must be mandatory for any biometric or identity system
- Define demographic performance thresholds **before** testing — not after seeing the results
- Real-world variation testing (lighting, accessories, aging) is functional testing, not an edge case
- The cost of building fairness in from the start is far lower than the cost of fixing it after launch

CASE STUDY 6: GPT-Based Customer Service Bot — Hallucination and Explainability Failures

Background

A major financial services company deployed a large language model-powered customer service assistant in 2023 to handle customer inquiries about products, account features, and regulatory information. The system was trained on product documentation and fine-tuned on historical customer service transcripts.

What Went Wrong

- The model began producing confident, fluent answers that were factually incorrect — a phenomenon known as **hallucination**
- It cited specific interest rates, fee structures, and regulatory terms that did not exist in the company's products
- Customers made financial decisions based on incorrect information provided by the bot
- When customers asked why they received incorrect information, there was no explanation mechanism — the model could not account for its outputs
- Monitoring was in place for customer satisfaction, but not for factual accuracy of responses
- The hallucination problem was not detected in testing because test cases used only "expected" queries — not the full range of real customer questions

The Testing Gap

No **hallucination testing** — no systematic testing of whether the model fabricated information

No **factual grounding verification** — no test to check whether outputs could be traced back to source documentation

No **confidence calibration testing** — the model expressed equal confidence whether it was correct or hallucinating

No **out-of-scope query testing** — no test of what the model did when asked questions outside its knowledge base

No **semantic accuracy monitoring** in production — only operational metrics (response time, CSAT) were monitored

GPT Customer Service Bot — Result & Lessons

The company faced regulatory scrutiny for providing incorrect financial information to customers. The bot was taken offline for a full remediation cycle including retrieval-augmented generation (RAG) implementation, source citation requirements, and human escalation for regulatory queries.

What Good AI Testing Would Have Caught

Ground Truth Testing

Comparing model answers against verified documentation for factual accuracy

Out-of-Scope Query Testing

What does the model do when asked something it should not answer?

Confidence Calibration

Is the model's expressed certainty aligned with its actual accuracy?

Adversarial Factual Testing

Deliberately asking about topics with subtle errors to see if the model corrects or amplifies them

Production Monitoring

Monitoring for factual accuracy using a second verification layer

Key Takeaways

- Fluency and confidence in AI output do not indicate accuracy — hallucination testing is non-negotiable for LLM deployments
- Monitoring must include output quality, not just operational metrics
- Financial, medical, and legal AI deployments require factual grounding verification before any output reaches users
- Out-of-scope behavior is a test case — what does the model do when it should not answer?
- RAG and source citation are architectural solutions — but they must be tested, not assumed to prevent hallucination

GSDC
Global Skill Development Council

CERTIFIED AI TESTING PROFESSIONAL (CAITP)

ABOUT GSDC CERTIFICATION

EBOOK

Extensive and exclusive Ebook created by world's experts to help you with understanding core concepts.

LEARNING MATERIALS

Get access to learning materials such as videos, ebooks, templates, and practice exams, which will help you clear the certification exam.

CREATED BY EXPERTS

GSDC certifications are created and authored by world's leading experts in the field.

LEARNING OBJECTIVE

- Gain insights into autonomous decision-making processes
- Apply knowledge using ready-to-implement templates
- Demonstrate ability to work with Agentic AI models
- Validate your skills wit

Enroll now with the code **LEARN20** To avail **20%** discount

Enroll Now

www.gsdcouncil.org