# **Silimate Copilot Product Analytics Project**

## **Executive Summary**

Build a data analytics system that measures the **true impact** of Silimate's chip design copilot—focusing on design intent adherence, engineering velocity, and product reliability rather than vanity metrics.

## **1. DATA ARCHITECTURE**

### **1.1 Core Data Sources**

#### **Design Session Data**

- session\_id (unique identifier)

- user\_id / engineer\_id

- customer\_id (company)

- project\_id (chip/IP being designed)

- session\_start\_time, session\_end\_time

- design\_stage (RTL, synthesis, PnR, verification, etc.)

- initial\_design\_state (snapshot)

- final\_design\_state (snapshot)

#### **Design Intent Metrics**

- target\_power (mW)

- target\_performance (MHz/GHz)

- target\_area (mm²)

- constraint\_violations (timing, power, area)

- distance\_from\_intent (% deviation from targets)

- intent\_achievement\_score (0-100)

#### **Copilot Interaction Data**

- interaction\_id

- session\_id

- timestamp

- interaction\_type (suggestion, fix, analysis, query)

- copilot\_suggestion (what it recommended)

- engineer\_action (accepted, modified, rejected, ignored)

- confidence\_score (AI model confidence 0-1)

- latency\_ms (response time)

#### **Design Change Events**

- change\_id

- session\_id

- timestamp

- change\_type (manual, copilot-assisted, copilot-auto)

- module\_modified

- lines\_of\_code\_changed

- change\_impact (functional, PPA optimization, bug fix)

- change\_success (did it improve metrics?)

#### **Error & Quality Data**

- error\_id

- session\_id

- timestamp

- error\_type (functional bug, PPA issue, constraint violation)

- error\_severity (critical, major, minor)

- detected\_by (copilot, engineer, verification tool)

- resolution\_time\_hours

- copilot\_suggested\_fix (yes/no)

- fix\_effectiveness (resolved, partial, ineffective)

#### **Performance Metrics**

- metric\_id

- session\_id

- timestamp

- power\_consumption (mW)

- clock\_frequency (MHz)

- chip\_area (mm²)

- timing\_violations\_count

- synthesis\_runtime\_hours

- verification\_coverage\_%

#### **User Behavior Data**

- feature\_usage\_id

- user\_id

- session\_id

- feature\_name

- time\_spent\_seconds

- abandonment\_event (switched to manual tool)

- abandonment\_reason (if captured)

#### **Customer & Project Metadata**

- customer\_id

- customer\_name

- customer\_tier (unicorn, enterprise, startup)

- chip\_type (SoC, IP block, ASIC)

- project\_complexity\_score

- team\_size

- contract\_value

- onboarding\_date

## **2. KEY METRICS FRAMEWORK**

### **2.1 Adoption & Growth Metrics**

#### **User Engagement**

* **Daily Active Users (DAU)** / **Weekly Active Users (WAU)**
* **Sessions per user per week**
* **Time spent in copilot vs. manual tools** (ratio)
* **Feature adoption rate** = % users who used feature X in last 30 days
* **Copilot assistance rate** = % of design changes made with copilot vs. manual

#### **Customer Growth**

* **Active customers** (used in last 30 days)
* **Projects using Silimate** (chip designs in progress)
* **Expansion within customers** (# of engineers using tool over time)
* **Stickiness** = DAU/MAU ratio

#### **Workflow Penetration**

* **Design stages covered** (which phases use copilot most: RTL, synthesis, PnR?)
* **Iteration cycles with copilot** (how many design iterations use tool)

### **2.2 Customer Satisfaction Metrics**

#### **Value Realization**

* **Time to first value** (days from onboarding to first successful PPA optimization)
* **Design velocity improvement** = (avg time per iteration WITH copilot) / (baseline manual time)
* **Design intent adherence score** = % reduction in constraint violations over time

#### **Friction Points**

* **Error rate** = errors per 100 copilot interactions
* **Rejection rate** = % of copilot suggestions rejected by engineers
* **Abandonment rate** = % of sessions where engineer switches back to manual tools
* **Latency frustration index** = % of interactions with >30s response time

#### **Trust Signals**

* **Suggestion acceptance rate** = % of copilot recommendations implemented
* **Repeat usage rate** = % of engineers who return to copilot within 7 days
* **Self-service success** = % of issues resolved without support escalation

#### **Support & Onboarding**

* **Time to proficiency** (days until engineer uses copilot independently)
* **Support ticket volume** (issues per 100 sessions)
* **Bug report frequency**

### **2.3 Product Quality Metrics**

#### **AI Model Performance**

* **Suggestion accuracy** = % of accepted suggestions that improved PPA
* **Hallucination rate** = % of suggestions that were technically incorrect
* **Confidence calibration** = correlation between model confidence and actual success
* **Prediction precision/recall** for bug detection

#### **System Performance**

* **P50, P95, P99 latency** (response time percentiles)
* **Availability** (uptime %)
* **Error rate per interaction**
* **Crash rate** (sessions ending in errors)

#### **Design Quality Impact**

* **PPA improvement rate** = % reduction in power/area, % increase in performance
* **Bug detection rate** = % of bugs caught by copilot vs. missed
* **Constraint violation reduction** = before/after copilot usage
* **Design intent convergence speed** = iterations needed to meet targets

#### **Release Quality**

* **Regression rate** = % of new releases introducing new bugs
* **Feature stability score** = % of features working without errors
* **Performance degradation** = latency increase across versions

### **2.4 Strategic Decision-Making Metrics**

#### **ROI Metrics**

* **Time savings per engineer** = hours saved per week using copilot
* **Cost per tapeout** = engineering hours × hourly cost (before vs. after)
* **Revenue per customer** correlated with product usage intensity

#### **Feature Impact**

* **Feature value score** = (usage frequency × satisfaction × PPA impact)
* **Feature ROI** = engineering effort invested vs. customer value delivered

#### **Customer Segmentation**

* **High-value customers** = usage frequency + contract value + NPS score
* **At-risk customers** = declining usage + high error rates + low satisfaction
* **Champion customers** = high adoption + successful outcomes (case study candidates)

#### **Roadmap Prioritization**

* **Feature request heatmap** = most requested features by customer tier
* **Bottleneck analysis** = design stages with highest friction
* **Competitive gaps** = features in traditional EDA tools that copilot lacks

## **3. DATA COLLECTION STRATEGY**

### **3.1 Instrumentation Plan**

#### **Event Tracking**

# Example event schema

{

"event\_type": "copilot\_suggestion",

"timestamp": "2025-09-30T14:23:15Z",

"session\_id": "sess\_abc123",

"user\_id": "eng\_456",

"customer\_id": "cust\_unicorn\_x",

"project\_id": "proj\_soc\_2024",

"context": {

"design\_stage": "synthesis",

"current\_ppa": {"power": 850, "freq": 2400, "area": 12.5},

"target\_ppa": {"power": 800, "freq": 2500, "area": 12.0}

},

"suggestion": {

"type": "optimization",

"description": "Reduce clock tree buffer stages",

"confidence": 0.87,

"expected\_impact": {"power": -30, "freq": 0, "area": -0.3}

},

"latency\_ms": 1240

}

#### **State Snapshots**

* Capture design state before/after copilot interactions
* Store PPA metrics at key checkpoints
* Log constraint violations continuously

#### **User Feedback**

* In-app thumbs up/down on suggestions
* Session-end surveys (optional, non-intrusive)
* NPS surveys quarterly

### **3.2 Data Pipeline**

[Copilot Tool] → [Event Streaming (Kafka/Kinesis)] → [Data Warehouse (Snowflake/BigQuery)]

↓

[Analytics Layer (dbt)]

↓

[Visualization (Looker/Tableau)]

## **4. ANALYSIS FRAMEWORKS**

### **4.1 Design Intent Adherence Analysis**

**Goal:** Measure how effectively copilot helps engineers reach their design targets.

**Metrics:**

Intent Achievement Score = 1 - (

|actual\_power - target\_power| / target\_power +

|actual\_freq - target\_freq| / target\_freq +

|actual\_area - target\_area| / target\_area

) / 3

Convergence Speed = iterations\_to\_meet\_constraints

**Analysis:**

* Track score improvement over project lifecycle
* Compare copilot-assisted vs. manual design phases
* Identify which constraints copilot optimizes best

### **4.2 Engineering Velocity Analysis**

**Goal:** Quantify time savings and productivity gains.

**Metrics:**

Time per Iteration = session\_duration / design\_changes\_made

Productivity Multiplier = baseline\_time / copilot\_assisted\_time

**Analysis:**

* Cohort analysis: compare teams using copilot vs. not
* Time-series: track velocity improvements over time
* Bottleneck identification: which design stages slow down most?

### **4.3 Error & Reliability Analysis**

**Goal:** Understand where copilot fails and why.

**Metrics:**

Error Rate = errors / total\_interactions

Hallucination Rate = incorrect\_suggestions / total\_suggestions

Recovery Time = time\_to\_fix\_copilot\_introduced\_errors

**Analysis:**

* Root cause analysis of rejected suggestions
* Error clustering (common failure patterns)
* Correlation: latency vs. error rate, confidence vs. accuracy

### **4.4 Customer Health Scoring**

**Goal:** Predict churn and identify expansion opportunities.

**Health Score Formula:**

Health Score = (

0.3 × Usage Score (sessions/week) +

0.3 × Satisfaction Score (NPS + acceptance rate) +

0.2 × Value Score (PPA improvements) +

0.2 × Stability Score (1 - error\_rate)

)

**Segments:**

* **Champions** (score >80): Expansion candidates, case studies
* **Healthy** (60-80): Stable, focus on retention
* **At-Risk** (<60): High-touch support, identify issues

## **5. DASHBOARD & REPORTING STRUCTURE**

### **5.1 Executive Dashboard (Weekly)**

* Active customers, users, projects
* Aggregate PPA improvements
* Customer health score distribution
* Top 5 friction points
* Revenue-correlated usage trends

### **5.2 Product Team Dashboard (Daily)**

* Feature usage trends
* Error rates by feature
* Latency P95 trends
* Suggestion acceptance rates
* Top bugs/feature requests

### **5.3 Customer Success Dashboard (Real-time)**

* Per-customer health scores
* Usage alerts (declining activity)
* Support ticket correlation
* Onboarding progress tracking

### **5.4 Engineering Dashboard (Real-time)**

* System performance metrics
* Error logs and stack traces
* Model performance (accuracy, latency)
* A/B test results

## **6. IMPLEMENTATION ROADMAP**

### **Phase 1: Foundation (Weeks 1-4)**

* [ ] Set up data warehouse and event streaming
* [ ] Implement core event tracking (sessions, interactions, errors)
* [ ] Build basic adoption dashboard (DAU, WAU, features used)

### **Phase 2: Quality Metrics (Weeks 5-8)**

* [ ] Add PPA tracking and design intent metrics
* [ ] Implement suggestion acceptance/rejection tracking
* [ ] Build error analysis pipeline

### **Phase 3: Advanced Analytics (Weeks 9-12)**

* [ ] Customer health scoring model
* [ ] Velocity impact analysis
* [ ] Predictive models (churn risk, expansion opportunity)

### **Phase 4: Automation & Alerts (Weeks 13-16)**

* [ ] Automated anomaly detection
* [ ] Real-time alerts (customer at-risk, critical errors)
* [ ] Self-serve analytics for product/sales teams

## **7. SUCCESS CRITERIA**

**This project succeeds if:**

1. **Product team can answer:** "Which feature should we build next?" with data
2. **Customer success can identify** at-risk customers 2 weeks before they churn
3. **Leadership can quantify** "Silimate reduces tapeout time by X months" with evidence
4. **Engineers can detect** performance regressions within 24 hours of a release
5. **Marketing can demonstrate** ROI to prospects with real customer data

## **8. TECHNICAL STACK RECOMMENDATIONS**

### **Data Infrastructure**

* **Event Streaming:** Apache Kafka or AWS Kinesis
* **Data Warehouse:** Snowflake (best for analytics) or BigQuery
* **ETL/Transformation:** dbt (data build tool)
* **Orchestration:** Airflow or Prefect

### **Analytics & Visualization**

* **BI Tool:** Looker, Tableau, or Metabase
* **Notebooks:** Jupyter + Python (pandas, scikit-learn)
* **Statistical Analysis:** Python (statsmodels, scipy) or R

### **Monitoring & Alerts**

* **Application Performance:** Datadog or New Relic
* **Custom Alerts:** PagerDuty integration
* **Logs:** ELK stack or Splunk

## **9. KEY RISKS & MITIGATIONS**

| **Risk** | **Mitigation** |
| --- | --- |
| **Proprietary data leakage** | Anonymize designs, hash identifiers, separate customer data |
| **Sampling bias** | Track non-usage (when engineers avoid copilot) |
| **Metric gaming** | Use composite scores, triangulate with qualitative feedback |
| **Data pipeline failures** | Monitoring, redundancy, automated tests |
| **Analysis paralysis** | Start with 5 core metrics, expand iteratively |

## **10. SAMPLE SQL QUERIES**

### **Design Intent Adherence Over Time**

WITH design\_checkpoints AS (

SELECT

session\_id,

user\_id,

customer\_id,

timestamp,

ABS(actual\_power - target\_power) / NULLIF(target\_power, 0) AS power\_error,

ABS(actual\_freq - target\_freq) / NULLIF(target\_freq, 0) AS freq\_error,

ABS(actual\_area - target\_area) / NULLIF(target\_area, 0) AS area\_error

FROM ppa\_metrics

)

SELECT

DATE\_TRUNC('week', timestamp) AS week,

customer\_id,

AVG(1 - (power\_error + freq\_error + area\_error) / 3) AS avg\_intent\_score

FROM design\_checkpoints

GROUP BY week, customer\_id

ORDER BY week DESC;

### **Suggestion Acceptance Rate by Feature**

SELECT

feature\_name,

COUNT(\*) AS total\_suggestions,

SUM(CASE WHEN engineer\_action = 'accepted' THEN 1 ELSE 0 END) AS accepted,

ROUND(100.0 \* accepted / total\_suggestions, 2) AS acceptance\_rate\_pct

FROM copilot\_interactions

WHERE interaction\_type = 'suggestion'

AND timestamp >= CURRENT\_DATE - INTERVAL '30 days'

GROUP BY feature\_name

ORDER BY acceptance\_rate\_pct DESC;

### **Customer Health Score**

WITH usage\_score AS (

SELECT customer\_id, AVG(sessions\_per\_week) / 20.0 AS score

FROM (

SELECT customer\_id, user\_id,

COUNT(DISTINCT session\_id) / 4.0 AS sessions\_per\_week

FROM sessions

WHERE timestamp >= CURRENT\_DATE - INTERVAL '28 days'

GROUP BY customer\_id, user\_id

)

GROUP BY customer\_id

),

satisfaction\_score AS (

SELECT customer\_id,

AVG(CASE WHEN engineer\_action = 'accepted' THEN 1.0 ELSE 0.0 END) AS score

FROM copilot\_interactions

WHERE timestamp >= CURRENT\_DATE - INTERVAL '28 days'

GROUP BY customer\_id

),

value\_score AS (

SELECT customer\_id,

AVG(intent\_achievement\_score) / 100.0 AS score

FROM design\_sessions

WHERE timestamp >= CURRENT\_DATE - INTERVAL '28 days'

GROUP BY customer\_id

)

SELECT

c.customer\_id,

c.customer\_name,

ROUND(

0.3 \* COALESCE(u.score, 0) +

0.3 \* COALESCE(s.score, 0) +

0.4 \* COALESCE(v.score, 0),

2

) \* 100 AS health\_score

FROM customers c

LEFT JOIN usage\_score u ON c.customer\_id = u.customer\_id

LEFT JOIN satisfaction\_score s ON c.customer\_id = s.customer\_id

LEFT JOIN value\_score v ON c.customer\_id = v.customer\_id

ORDER BY health\_score DESC;

## **NEXT STEPS**

1. **Define data collection spec** with engineering team
2. **Set up tracking** for 5 core events (sessions, suggestions, errors, PPA, abandonment)
3. **Build MVP dashboard** with adoption + quality metrics
4. **Validate metrics** with 2-3 pilot customers
5. **Iterate based on feedback** from product/CS/leadership teams

**Timeline:** 12-16 weeks to full implementation  
 **Resources Needed:** 1 data engineer, 1 analytics engineer, product manager input