



Turning Data into Action: Front-End Analytics for Smarter Products and Engineering Growth

Smarter Decisions, Better UX, Faster Engineering



Raju Dandigam

Engineering Manager and Staff Engineer at Navan

What does Front-End Analytics mean?



Real-Time Tracking

Tracking user interactions directly in the browser as they happen.



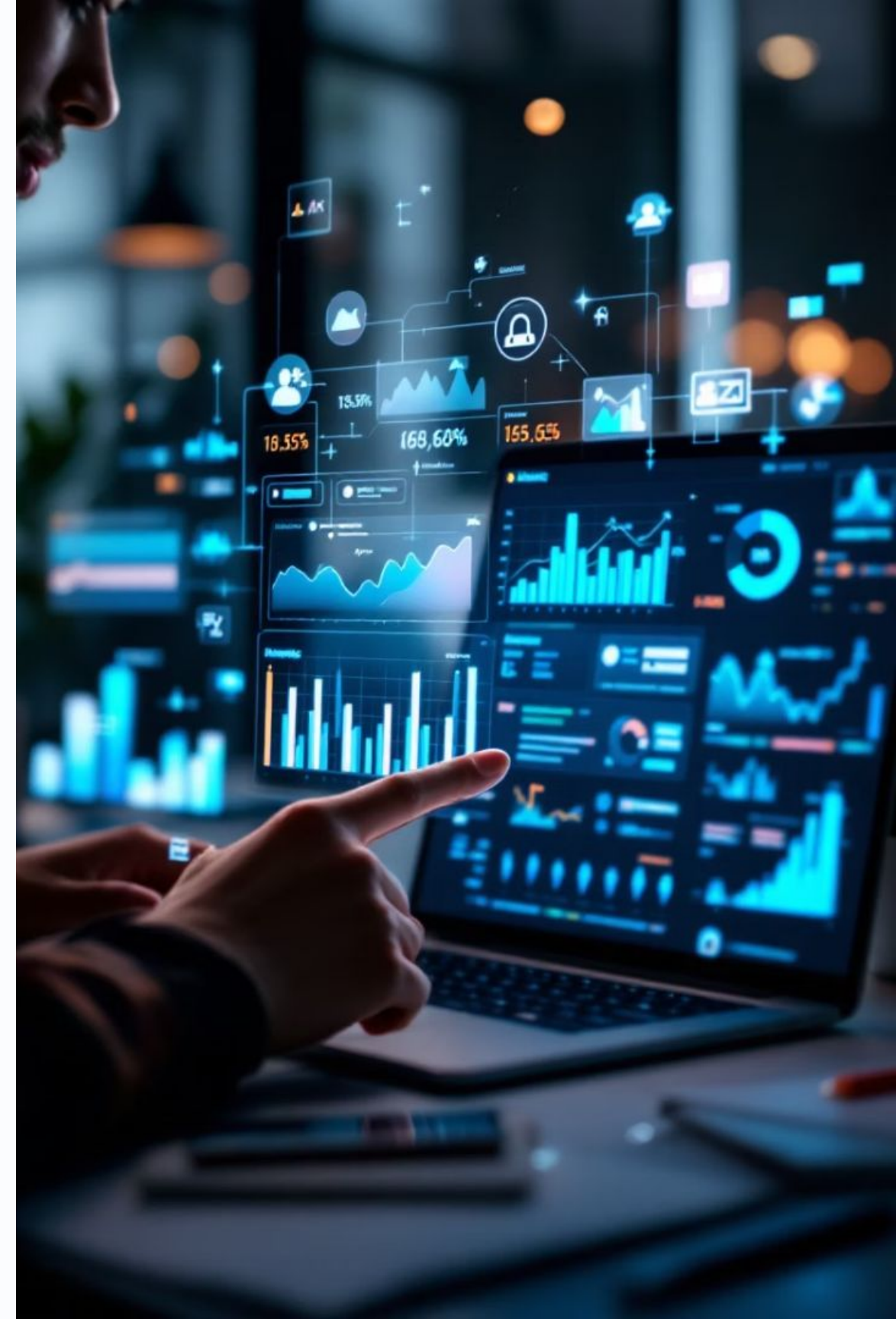
Actual Behavior

Observing what users actually do, not just what they say.

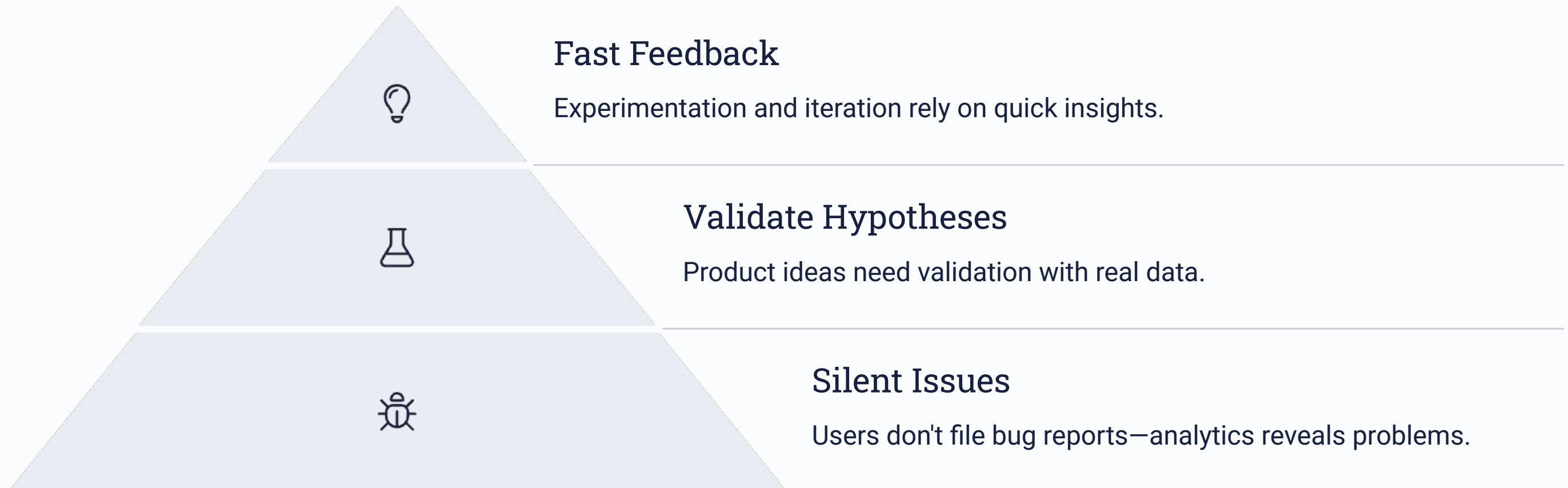


Interaction Data

Forms, clicks, pageviews, rage clicks, session time, abandons, scrolls.



Why It Matters



"Without data, you're just another person with an opinion." — W. Edwards Deming



Who Benefits?

Team	Use Case
Product	Feature prioritization, UX testing, Funnel analysis
Engineering	Error debugging, performance optimization
Growth	A/B testing, conversion funnels
Design	UI/UX validation, heatmaps, Experiments



What Can You Track?

Custom Events

Button clicks, searches, form submissions.



Performance

Page load times, scroll depth, time on screen.



User Flows


From landing → action → exit.

Pro Tip: Track *intent*, not just interactions.

Types of Front-End Analytics

Purpose	Examples
Behavioral	Funnel, retention, user journey
Technical	JS errors, performance metrics
Experimental	A/B Tests, Feature Flags
Session Replay	User recording (FullStory/OpenReplay)





Popular Tools: A Quick Overview

Tool	Type	Self-hosted
Amplitude	Behavioral/Product analytics	✗
PostHog	Behavioral + Replay capabilities	✓
Sentry	Error monitoring focus	✓
Mixpanel & LogRocket	Product analytics / Session replay + errors	✗

"For this session, I selected Amplitude as the example. "

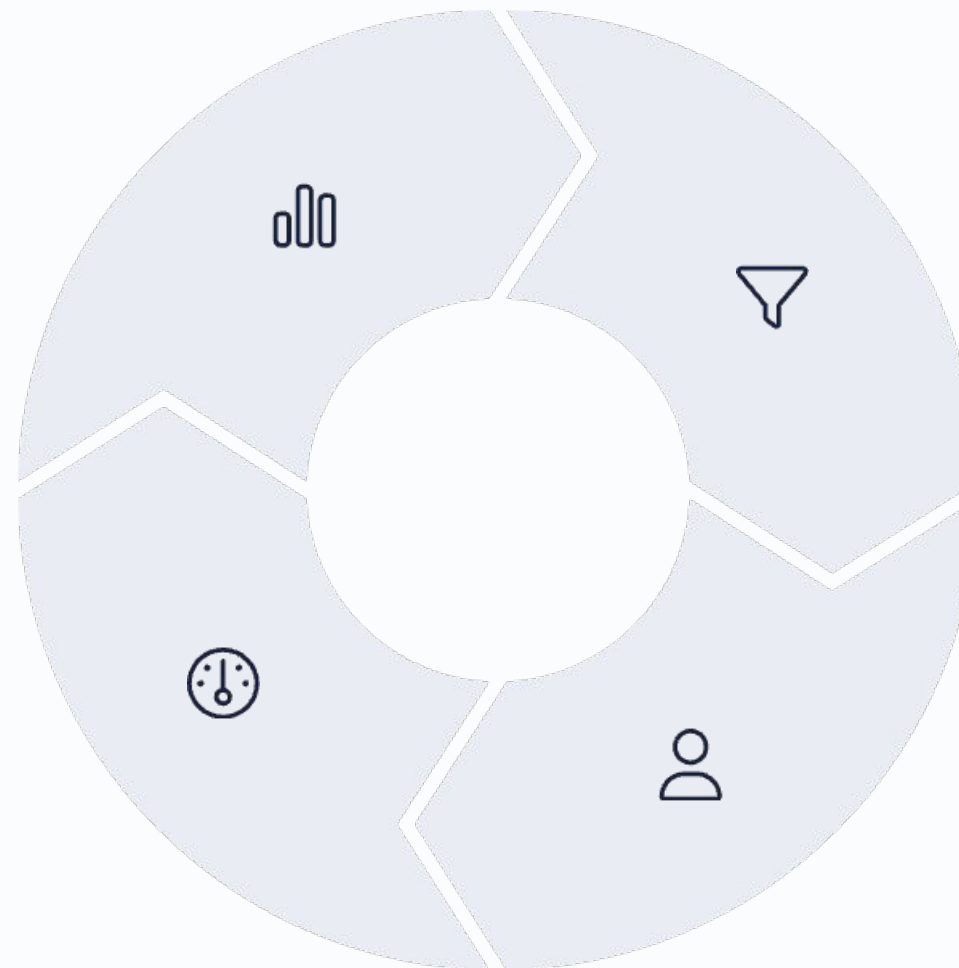
Why Amplitude?

Product-Focused

Built for event tracking that matters to product teams.

Dashboards

Powerful visualizations with easy team sharing.



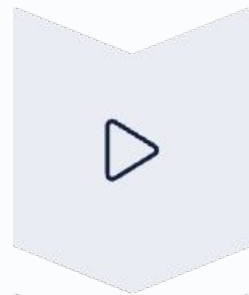
Powerful Analysis

Beautiful funnel + cohort analysis tools.

Collaboration

Great for engineering-product teamwork.

Real-World Example: Signup Funnel



signup_start

User begins the signup process.



email_entered

User provides their email address.



signup_success

User completes sign up successfully.

Use Funnel chart to see drop-offs.

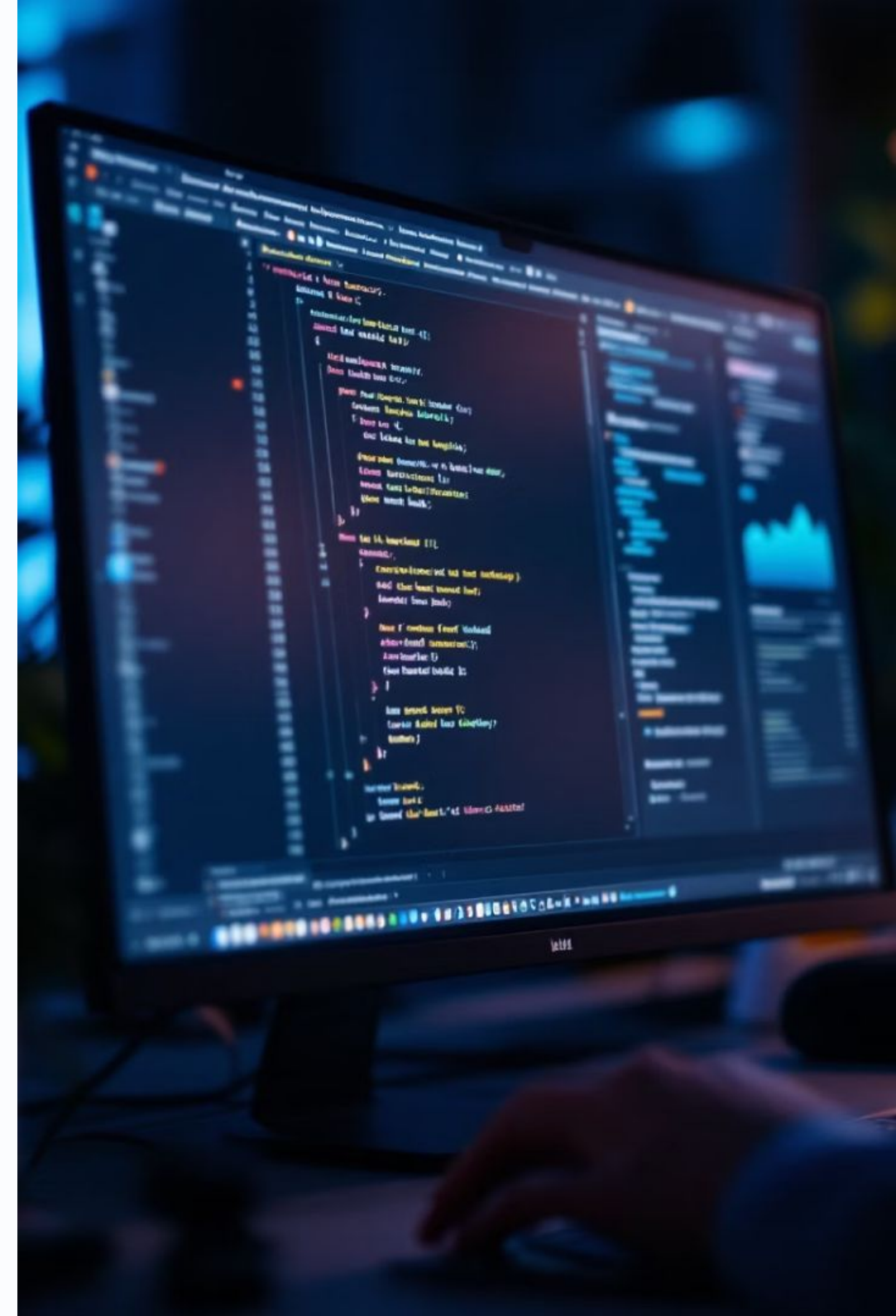
Run experiment: shorter form vs. long form.



How to Instrument Amplitude

```
// index.js
import amplitude from '@amplitude/analytics-browser';
amplitude.init('YOUR_API_KEY');

// App.js
import React from 'react';
import amplitude from '@amplitude/analytics-browser';
function App() {
  const handleClick = () => {
    amplitude.logEvent('Button Clicked', { buttonName: 'testButton' });
  };
  return (<div> <button onClick={handleClick}>Click</button> </div>);
}
export default App;
```



Optimizing Your Analytics Setup

Predefine Event Taxonomy

Create a consistent structure before implementation.

Use Consistent Naming

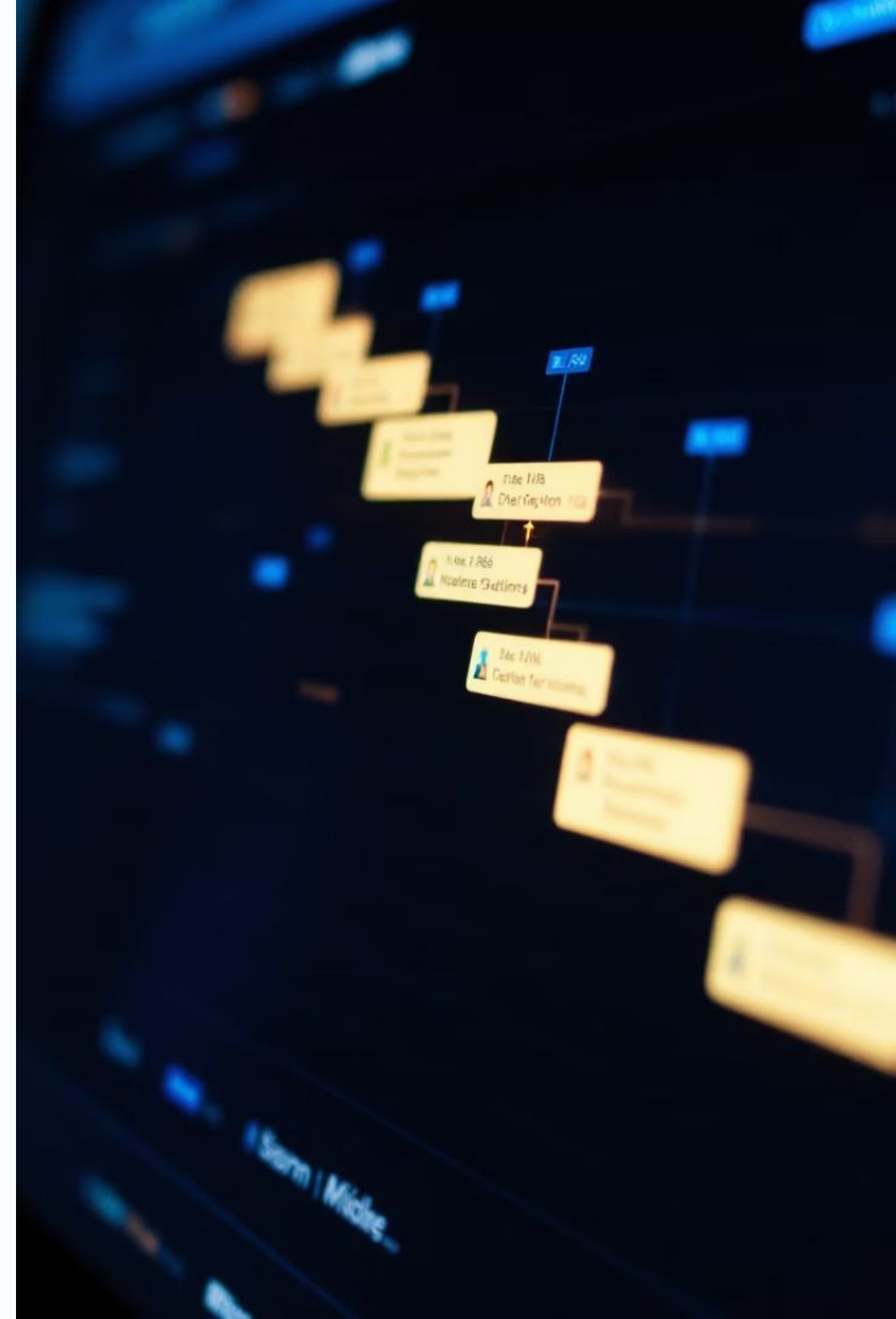
Stick with snake_case for all events.

Limit Properties

Track only key actions to avoid data overload.

Track User ID

Use `amplitude.setUserId()` after authentication.



Running A/B Tests



Assign Users

Randomly distribute users to variants.



Track Events

Log `experiment_viewed` and `experiment_converted`.



Measure Lift

View conversion improvement over baseline.

Use Amplitude Experiment or integrate with external A/B testing tools for comprehensive testing.



Debugging with Analytics



Identify Drop

Notice unexpected decrease in checkout_started events.



Check Deployments

Look for correlation with recent code changes.



Segment Data

Break down by browser or device to isolate issues.



Review Sessions

Watch user recordings if integration exists.

Tie analytics into your CI/CD loop or QA dashboard for proactive monitoring.

Key Takeaways

1

Feedback Loop

Front-end analytics is more than metrics—it's a continuous feedback system.

2

Data-Driven

Amplitude enables fast, evidence-based decision-making.

3

Start Small

Begin with 3–5 events to create meaningful dashboards.

4

Team Win

Engineering and product teams succeed together with shared insights.

