Dashboard
Renaissance
Clever Subtitle Here
Hello, I’m Cory

- Technical Evangelist @ Splunk
- previously:
  - Director, Office of the CTO @ SignalFx
  - Principal Engineer, Stripe
  - SRE & Engineering Manager, Twitter
- observability wonk
Goal: Improve dashboards

- Humans are what matter
- Often the “UI” for our work as operators
- We’re not trained to do this
- Impacts on employees and customers
- Cognitive resources are shared
“Mistakes often arise from ambiguous or unclear information about the current state of the system”

- Don Norman, *The Design of Everyday Things*
Don Norman’s Mistakes

- Mistakes: conscious decision of what to do, gone wrong
- “Poor information provided by the system”
- “Poor-quality feedback about what has actually happened”
Metrics & Layout
Get a spec from users

- Purpose
- Goal
- Familiarity
- Context needed
- Actions
4 Types of Dashboard

- **Steady-state**
  - latency, errors, txn counts, outcomes
- **Long-term**
  - planning, scaling, big time windows
- **Diagnostic**
  - what’s broken? blast radius, hypotheses
- **Debugging**
  - lots of guessing, disproving, ad hoc
Valuable Metrics

- Understandable
- Comparable (business, time, resource, customers)
- Drives action or change
- Unavailable elsewhere
Starting Concepts

● Use your goals and priorities
● Grids are good
What to include

- RED and USE
- Be choosey
- Prefer symptoms to causes
- Meaningful comparisons
One Screen, More Next Steps

- Colocate common items
- Humans can only keep 3 or 4 visuals in working memory

<table>
<thead>
<tr>
<th>Overview</th>
<th>HTTP Details</th>
<th>Dependencies</th>
<th>Networking</th>
<th>Runtime Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>First thing</td>
<td>Compare A</td>
<td>Compare B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Next logical thing</td>
<td>... A</td>
<td>... B</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Progressive Depth

Dashboard 1
Overview (KPI)

Dashboard 2
HTTP Detail

Dashboard 3
Deps / Net

Dashboard 4
Runtime / OS

Failover, mitigation, etc

Other dashboards, mesh diagnostics

Other teams, logs

SSH to the darn box

Check out Ecological Interface Design!
Summary: Design It!

- Get spec from users
- Include metrics that meet goals
- Use grid
- Support progression, linking to more things
Visualizations
Hacking Humans

- What can we quickly get information from?
- What is pleasing to the eye?
- How can we increase accuracy?
- Where do we draw attention?
Preattentive Processing

089762345762334379561337509
272345098723409857092384752
136ENGAGE9WITH8MY4BRAND1567
Preattentive Processing
Preattentive Processing

089762345762334379561337509
272345098723409857092384752
136ENGAGE9WITH8MY4BRAND1567
Preattentive Processing
Preattentive Processing

- Intensity
- Orientation
- Length
- Shape
- Enclosure
- Hue
- Position
Accuracy Of Perception
Humans Like Horizons

- Wider than taller
- Time as X axis is intuitive
- Angles of data are better (fewer tall spikes)
Before Lines, Bars, & etc

- Alerts are just charts you care about
- Position and meaning
Don’t Forget Tables

- Good for single values
- Well understood, dense

<table>
<thead>
<tr>
<th>Value</th>
<th>demo_host</th>
<th>demo_customer</th>
<th>demo_data</th>
<th>Plot Name</th>
<th>sf_metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>205.47ms</td>
<td>server2</td>
<td>zibobodesign.net</td>
<td>Paris</td>
<td>Operation Duration</td>
<td>demo.trans.latency</td>
</tr>
<tr>
<td>205.13ms</td>
<td>server2</td>
<td>thefountain.org</td>
<td>Paris</td>
<td>Operation Duration</td>
<td>demo.trans.latency</td>
</tr>
<tr>
<td>204.38ms</td>
<td>server3</td>
<td>thefountain.org</td>
<td>Paris</td>
<td>Operation Duration</td>
<td>demo.trans.latency</td>
</tr>
<tr>
<td>204.37ms</td>
<td>server4</td>
<td>thefountain.org</td>
<td>Paris</td>
<td>Operation Duration</td>
<td>demo.trans.latency</td>
</tr>
<tr>
<td>204.26ms</td>
<td>server4</td>
<td>zibobodesign.net</td>
<td>Tokyo</td>
<td>Operation Duration</td>
<td>demo.trans.latency</td>
</tr>
<tr>
<td>204.19ms</td>
<td>server3</td>
<td>zibobodesign.net</td>
<td>Tokyo</td>
<td>Operation Duration</td>
<td>demo.trans.latency</td>
</tr>
<tr>
<td>203.34ms</td>
<td>server6</td>
<td>zibobodesign.net</td>
<td>Tokyo</td>
<td>Operation Duration</td>
<td>demo.trans.latency</td>
</tr>
<tr>
<td>202.16ms</td>
<td>server6</td>
<td>thefountain.org</td>
<td>Tokyo</td>
<td>Operation Duration</td>
<td>demo.trans.latency</td>
</tr>
<tr>
<td>202.13ms</td>
<td>server1</td>
<td>zibobodesign.net</td>
<td>Paris</td>
<td>Operation Duration</td>
<td>demo.trans.latency</td>
</tr>
<tr>
<td>202.16ms</td>
<td>server5</td>
<td>thefountain.org</td>
<td>Tokyo</td>
<td>Operation Duration</td>
<td>demo.trans.latency</td>
</tr>
<tr>
<td>201.28ms</td>
<td>server5</td>
<td>zibobodesign.net</td>
<td>Tokyo</td>
<td>Operation Duration</td>
<td>demo.trans.latency</td>
</tr>
<tr>
<td>201.15ms</td>
<td>server2</td>
<td>samslack.com</td>
<td>Paris</td>
<td>Operation Duration</td>
<td>demo.trans.latency</td>
</tr>
<tr>
<td>200.29ms</td>
<td>server4</td>
<td>samslack.com</td>
<td>Tokyo</td>
<td>Operation Duration</td>
<td>demo.trans.latency</td>
</tr>
<tr>
<td>200.14ms</td>
<td>server3</td>
<td>samslack.com</td>
<td>Paris</td>
<td>Operation Duration</td>
<td>demo.trans.latency</td>
</tr>
<tr>
<td>199.12ms</td>
<td>server6</td>
<td>samslack.com</td>
<td>Tokyo</td>
<td>Operation Duration</td>
<td>demo.trans.latency</td>
</tr>
<tr>
<td>198ms</td>
<td>server1</td>
<td>samslack.com</td>
<td>Paris</td>
<td>Operation Duration</td>
<td>demo.trans.latency</td>
</tr>
<tr>
<td>197.27ms</td>
<td>server5</td>
<td>samslack.com</td>
<td>Tokyo</td>
<td>Operation Duration</td>
<td>demo.trans.latency</td>
</tr>
</tbody>
</table>
Tables Can Be Fancy

- Just the fields you want
- Sparklines, etc

<table>
<thead>
<tr>
<th>Current Latency By Customer</th>
<th>10s</th>
<th>Current Latency By DC, Customer</th>
<th>10s</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>200ms SLO</td>
<td></td>
</tr>
<tr>
<td>Paris</td>
<td>zibobodesign.net</td>
<td>247.2ms</td>
<td>Tokyo</td>
</tr>
<tr>
<td>Paris</td>
<td>thefountain.org</td>
<td>247.1ms</td>
<td>Tokyo</td>
</tr>
<tr>
<td>Tokyo</td>
<td>zibobodesign.net</td>
<td>213.2ms</td>
<td>Tokyo</td>
</tr>
<tr>
<td>Tokyo</td>
<td>samslack.com</td>
<td>209.2ms</td>
<td>Tokyo</td>
</tr>
<tr>
<td>Tokyo</td>
<td>samslack.com</td>
<td>242.2ms</td>
<td>Tokyo</td>
</tr>
<tr>
<td>Tokyo</td>
<td>thefountain.org</td>
<td>213.5ms</td>
<td>Paris</td>
</tr>
</tbody>
</table>
Line Versus Heat

- Too many lines actually hide signal
- Heatmaps show outliers
- Saturation is low accuracy tho
Area?

- Good for “parts of a sum”
- **Only** use stacked
- But which things are contributing?
(Sometimes) Better Than Area

- Show sum as single line
- Separate per-item unstacked line for outliers
Bars

- Beware aggregation
- Best for comparisons of a few values
Sneaky Aggregation

- Beware aggregation
- Very common source of confusion
Other Junk

- Don’t use gauges with no upper & lower bounds
- Graphics not conveying data waste of pixels
- Beware of too many colors, alerts, and distractions
Visualization Choices

● Use tables for single values
● Use lines for $1 > x < a \ few$
● **Seriously, use line charts**
● Heat maps for lots of lines
● Bars for small comparisons
Context
Practitioners Need Context

- We need to empower humans
- There’s lots of context we can add
  - Info about the data
  - Happenings in the org
  - Next steps
Scale, Units, Norms, Labels
Bounds

API Call Duration

Duration of API call in milliseconds as measured from middleware

Call Duration

Timeout

SLO Expectation

0

13:45  13:50  13:55  14:00  14:05  14:10

Max

Expectation

Current!

Min
● Stop being lazy, write some stuff
● Descriptions, titles, watermarks, etc
Mark Targets

- Facilitates comparison of data to goal
- Much less cognitive load
More Text

- Help out the user, give links and explanations
- What does the user do with this?

Charts - Trend
Total number of charts

Dashboards - Trend
Total number of dashboards

Charts - Info
Metric: sf.org.num.chart
Total number of charts; includes any charts created using the SignalFx API but not associated with a dashboard.

Dashboards - Info
Metric: sf.org.num.dashboard
Total number of dashboards; includes all user, custom, and built-in dashboards.
Next Steps

● What can the user do?
● Iterate and improve this!
Situational Awareness

- Control-rate events
- Deploys, feature flags, scaling, etc
- Make this easy to toggle

![Feedback PID Controller - Output And Value](chart)

<table>
<thead>
<tr>
<th>DATA TABLE</th>
<th>EVENTS (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timestamp</td>
<td>Description</td>
</tr>
<tr>
<td>Aug 21, 2019 9:06:14 AM</td>
<td>Cory deployed ad2ae85</td>
</tr>
</tbody>
</table>
Dashboard with important things, careful attention grabbers, and context

[Diagram showing important and supporting elements with some highlighted as critical]
Charts with clear explanation, bounds, and next steps
Why do this?

- Tools only useful if used by humans
- This is how we manage complexity
- Decreasing cognitive effort leaves more room for knowledge-based work
How to demonstrate value

- Measure usage
- Solicit feedback
- Train
- Use with deployments, chaos/fault inj/gameday
- Collect usage with incidents
- Garden
- Consider the whole
- Integration with tools
How to revive your dashboards

- Deliberately design
- Support progression
- Prefer symptoms, use tables and line charts
- Emphasize what’s important, favor next steps
- For crying out loud label, describe, and link
- Add context, bruh
Grow capabilities

- Eliminate unnecessary work
- Balance automation
- Teach up the basics
- Encourage experiments and practice
Would you like to know more?

- Information Dashboard Design
  Stephen Few

- The Design of Everyday Things
  Don Norman

- The Visual Display of Quantitative Information
  Edward R. Tufte

- usability.gov
  Department of Health & Human Services
Thank you!

- @gphat
- onemogin.com
- Thanks to SignalFx/Splunk for paying me
- Come see me!