UX Design and Education for Effective Monitoring Tools
Agenda

- Background
- Motivation
- Sharing what you know
- Designing what your users want
- Recap
Agenda

- Background
- Motivation
- Sharing what you know
- Designing what your users want
- Recap
Hi!👋

- Joined Pinterest in 2015
Hi!👋

- Joined Pinterest in 2015
- Recent projects:
Hi!👋

- Joined Pinterest in 2015
- Recent projects:
  o tracing for performance
Hi!

- Joined Pinterest in 2015
- Recent projects:
  ○ tracing for performance
  ○ D3 data visualizations
Hi!

- Joined Pinterest in 2015

- Recent projects:
  ○ tracing for performance
  ○ D3 data visualizations
  ○ cache for OpenTSDB data
Hi!

- Joined Pinterest in 2015
- Recent projects:
  - tracing for performance
  - D3 data visualizations
  - cache for OpenTSDB data
  - *so much documentation omg*
Hi!

- Joined Pinterest in 2015
- Recent projects:
  ○ tracing for performance
  ○ D3 data visualizations
  ○ cache for OpenTSDB data
  ○ so much documentation omg
- amynguyen.net
  @amyngyn
  pinterest.com/amyngyn
About Pinterest

the world's first visual discovery engine
About Pinterest

*the world's first visual discovery engine*

- 175 million monthly active users
About Pinterest

*the world's first visual discovery engine*

- 175 million monthly active users
- 100 billion pins
About Pinterest

the world's first visual discovery engine

- 175 million monthly active users
- 100 billion pins
- 2 billion boards
About Pinterest

the world's first visual discovery engine

- 175 million monthly active users
- 100 billion pins
- 2 billion boards
- 2 billion searches every month
About Pinterest

the world's first visual discovery engine

- 175 million monthly active users
- 100 billion pins
- 2 billion boards
- 2 billion searches every month
- 150,000 requests served per second
About Pinterest Monitoring

the world's first greatest monitoring team
About Pinterest Monitoring

*the world's first greatest monitoring team*

- Graphite, OpenTSDB, Kafka, Storm, Spark, ELK, Sumo Logic, Zipkin
About Pinterest Monitoring

*the world's first greatest monitoring team*

- Graphite, OpenTSDB, Kafka, Storm, Spark, ELK, Sumo Logic, Zipkin
- 100 terabytes logged per day
About Pinterest Monitoring

*the world's first greatest monitoring team*

- Graphite, OpenTSDB, Kafka, Storm, Spark, ELK, Sumo Logic, Zipkin
- 100 terabytes logged per day
- 2.5M metrics ingested per second
About Pinterest Monitoring

the world's first greatest monitoring team

- Graphite, OpenTSDB, Kafka, Storm, Spark, ELK, Sumo Logic, Zipkin
- 100 terabytes logged per day
- 2.5M metrics ingested per second
- Over 400 engineers
About Pinterest Monitoring

*the world's first greatest monitoring team*

- Graphite, OpenTSDB, Kafka, Storm, Spark, ELK, Sumo Logic, Zipkin
- 100 terabytes logged per day
- 2.5M metrics ingested per second
- Over 400 engineers customers!
Our Tools: Dashboards
Our Tools: Graph Exploration
Our Tools: Alerting

Sample Alert

maxSeries(redacted_metric) > 0.07

When

From 10 minutes ago until now

Aggregate series by average

Check every 30 seconds

For empty data, treat as passing

For graphite metrics, replace null with

Test Alert
Agenda

- Background

- Motivation
  - Sharing what you know
  - Designing what your users want

- Recap
Why should we care about user experience?
Why should we care about user experience?
Why should we care about user experience?

- **Prevent misunderstandings:** not everyone is (or should have to be) an expert at interpreting monitoring data
Why should we care about user experience?

- **Prevent misunderstandings:** not everyone is (or should have to be) an expert at interpreting monitoring data

- **Developer velocity:** help people reach conclusions faster, help your company move faster
Why should we care about user experience?

- **Prevent misunderstandings:** not everyone is (or should have to be) an expert at interpreting monitoring data

- **Developer velocity:** help people reach conclusions faster, help your company move faster

- **Data democracy:** you don’t know what questions people want to answer with their own data
"The fastest way to become a 10x engineer is to help 10 other engineers do their jobs better.

- Wayne Gretzky”

- Michael Scott
Why should we care about user experience?

- **Prevent misunderstandings**: not everyone is (or should have to be) an expert at interpreting timeseries data

- **Developer velocity**: help people reach conclusions faster, help your company move faster

- **Data democracy**: you don’t know what questions people want to answer with their own data
Why should we care about user experience?

- **Prevent misunderstandings:** not everyone is (or should have to be) an expert at interpreting timeseries data.

- **Developer velocity:** help people reach conclusions faster, help your company move faster.

- **Data democracy:** you don’t know what questions people want to answer with their own data.
Why should we care about user experience?

Because we can help engineers work correctly, quickly, and independently.
UX and Your Situation
UX and Your Situation
UX and Your Situation

documentation

team

Amy Nguyen  @amyngyn  Monitorama PDX 2017
UX and Your Situation

- team
- documentation
- tools
UX and Your Situation

- team
- documentation
- tools

probably paying for vendors TBH?
UX and Your Situation

this talk

tools

documentation

team

probably paying for vendors TBH?
UX and Your Situation

documentation

tools

this talk

things we can all control

probably paying for vendors TBH?
Agenda

- Background
- Motivation
- Sharing what you know
- Designing what your users want
- Recap
Sharing what you know
Sharing what you know

1. **Education vs intuition:** Don’t overload people with too much information.
will the wifi work? who knows?
stats.example.metric.errors > 5
Statsboard FAQ

Created by Amy Nguyen, last modified 49 minutes ago

- Exploring Metrics
  - The difference between Statsboard, OpenTSDB, Graphite, and Logsearch
  - OpenTSDB aggregators (sum:metric, avg:metric)
  - Searching for metrics
  - Tag value filtering
- Creating metrics
  - Tcollector
  - Python
  - Java
  - Node
- Dashboards
  - Creation
  - Previewing
  - Configuration
  - Composite dashboards
Sharing what you know

1. **Education vs intuition**: Don’t overload people with too much information.

2. **Best practices**: Use your expertise to determine the most helpful default behavior.
maxSeries(redacted_metric) > 0.7

- From 10 minutes ago until now
- Aggregate series by average
- Check every 30 seconds
- For empty data: do nothing
- For graphite metrics, replace null with

Test Alert
Sharing what you know

1. **Education vs intuition:** Don’t overload people with too much information.

2. **Best practices:** Use your expertise to determine the most helpful default behavior.

3. **Potential pitfalls:** Make it hard to do the wrong thing.
maxSeries(redacted_metric) > 0.7

When

From 10 minutes ago until now

Aggregate series by final point

Check every 30 minutes

Drop edges

For empty data, do nothing

For graphite metrics, replace null with
It looks like you're trying to alert on the most recent data. Are you sure you want to do that?
Agenda

- Background
- Motivation
- Sharing what you know
- Designing what your users want
- Recap
Designing what your users want

1. **Performance**: Do whatever it takes to make it fast.
Performance: Low Hanging Fruit!
Performance: Low Hanging Fruit!

- Backend
Performance: Low Hanging Fruit!

- Backend
  - Roll-up data over long time ranges
Performance: Low Hanging Fruit!

- Backend
  - Roll-up data over long time ranges
  - Store latest data in memory (e.g., Facebook’s [Gorilla paper](http://example.com) and [Beringei project](http://example.com))
Performance: Low Hanging Fruit!

- **Backend**
  - Roll-up data over long time ranges
  - Store latest data in memory (e.g., Facebook’s [Gorilla paper](#) and [Beringei project](#))
  - Add a cache layer (e.g., [Turn’s Splicer project](#))
Performance: Low Hanging Fruit!

- Backend
  - Roll-up data over long time ranges
  - Store latest data in memory (e.g., Facebook’s Gorilla paper and Beringei project)
  - Add a cache layer (e.g., Turn’s Splicer project)

yeah sure get back to us in 6 months
Performance: Low Hanging Fruit!

- **Backend**
  - Roll-up data over long time ranges
  - Store latest data in memory (e.g., Facebook’s [Gorilla paper](#) and [Beringei project](#))
  - Add a cache layer (e.g., [Turn’s Splicer project](#))

- **Frontend**
Performance: Low Hanging Fruit!

- **Backend**
  - Roll-up data over long time ranges
  - Store latest data in memory (e.g., Facebook’s [Gorilla paper](https://www.facebook.com/notes/mustang/monitoring-data-at-facebook/) and [Beringei project](https://github.com/beringei/beringei))
  - Add a cache layer (e.g., Turn’s [Splicer project](https://github.com/turnernet/splicer))

- **Frontend**
  - Don’t reload existing data if the user changes the time window
Performance: Low Hanging Fruit!

- **Backend**
  - Roll-up data over long time ranges
  - Store latest data in memory (e.g., Facebook’s Gorilla paper and Beringei project)
  - Add a cache layer (e.g., Turn’s Splicer project)

- **Frontend**
  - Don’t reload existing data if the user changes the time window
  - Prevent the user from requesting the data incessantly
Performance: Low Hanging Fruit!

- **Backend**
  - Roll-up data over long time ranges
  - Store latest data in memory (e.g., Facebook’s Gorilla paper and Beringei project)
  - Add a cache layer (e.g., Turn’s Splicer project)

- **Frontend**
  - Don’t reload existing data if the user changes the time window
  - Prevent the user from requesting the data incessantly
  - Lazy-load graphs on a dashboard
Performance: Low Hanging Fruit!

- **Backend**
  - Roll-up data over long time ranges
  - Store latest data in memory (e.g., Facebook’s [Gorilla paper](#) and [Beringei project](#))
  - Add a cache layer (e.g., [Turn’s Splicer project](#))

- **Frontend**
  - Don’t reload existing data if the user changes the time window
  - Prevent the user from requesting the data incessantly
  - Lazy-load graphs on a dashboard

- **Disclaimer:** We haven’t done all of these things.
Designing what your users want

1. **Performance**: Do whatever it takes to make it fast.

2. **Exploration**: Make it easy to try things without fear.
Greatest Graph of All Time

Click and drag to zoom. Right click to zoom out.

Metrics | Statistics | Raw | Display | YAML
--- | --- | --- | --- | ---
stats:relected.metric:foo

- example_tag_key

sum:stats:relected.metric:bar

- example_tag_key

sum:stats:relected.metric:baz

- example_tag_key

Amy Nguyen
@amyngyn
Monitorama PDX 2017
Designing what your users want

1. **Performance**: Do whatever it takes to make it fast.

2. **Exploration**: Make it easy to try things without fear.

3. **Simplicity**: Make it easy to figure out what to do.
Am I so out of touch?
sum:10m-avg:stats.example.metric.errors{host=my_host, other_tag=some_value}
manually type in your metric if you know the name somehow

```
sum:10m-avg:stats.example.metric.errors{host=my_host, other_tag=some_value}
```
manually type in your metric if you know the name somehow

sum:10m-avg:stats.example.metric.errors{host=my_host, other_tag=some_value}
manually type in your metric if you know the name somehow

textbox:
```
sum:10m-avg:stats.example.metric.errors{host=my_host, other_tag=some_value}
```

aggregator

hope you remember what tags are available for this metric lol
manually type in your metric if you know the name somehow

secret bonus: you can downsample?!

aggregator

hope you remember what tags are available for this metric lol
manually type in your metric if you know the name somehow

aggregator

secret bonus: you can downsample?!

hope you remember what tags are available for this metric lol

sum:10m-avg:stats.example.metric.errors{host=my_host, other_tag=some_value}
<table>
<thead>
<tr>
<th>Metrics</th>
<th>Statistics</th>
<th>Raw</th>
<th>Display</th>
<th>YAML</th>
</tr>
</thead>
</table>

`stat.redacted.metric`

**My Example Metric**

- OpenTSDB: `sum`
- `Rate` | `Downsample`

**Grouping Tags**

- go
- host
- foo
- joinPartition
- partition
- stage

**Non-grouping Tags**

- service
- bar
- topic
- baz
- version
- qux

Amy Nguyen
@amyngyn
Monitorama PDX 2017
### My Example Metric

<table>
<thead>
<tr>
<th>Grouping Tags</th>
<th>Host</th>
<th>Foo</th>
<th>JoinPartition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Go</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Partition</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Non-grouping Tags</th>
<th>Service</th>
<th>Bar</th>
<th>Topic</th>
<th>Baz</th>
<th>Version</th>
<th>Qux</th>
</tr>
</thead>
</table>

Available options you probably don’t need to touch.
<table>
<thead>
<tr>
<th>Grouping Tags</th>
<th>Non-grouping Tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>go</td>
<td>service</td>
</tr>
<tr>
<td>host</td>
<td>bar</td>
</tr>
<tr>
<td>foo</td>
<td>topic</td>
</tr>
<tr>
<td>joinPartition</td>
<td>baz</td>
</tr>
<tr>
<td>stage</td>
<td>version</td>
</tr>
<tr>
<td></td>
<td>qux</td>
</tr>
</tbody>
</table>

manual entry for power users

stat.redacted.metric
relevant information needed to create a query!
Designing what your users want

1. **Performance**: Do whatever it takes to make it fast.

2. **Exploration**: Make it easy to try things without fear.

3. **Simplicity**: Make it easy to figure out what to do.
Agenda

- Background
- Motivation
- Sharing what you know
- Designing what your users want
- Recap
"The fastest way to become a 10x engineer is to help 10 other engineers do their jobs better.

- Wayne Gretzky"

- Michael Scott
UX and Your Situation

this talk

things we can all control

probably paying for vendors TBH?

tools

documentation

team

Amy Nguyen
@amyngyn
Monitorama PDX 2017
Sharing what you know

1. **Education vs intuition**: Don’t overload people with too much information.

2. **Best practices**: Use your expertise to determine the most helpful default behavior.

3. **Potential pitfalls**: Make it hard to do the wrong thing.
Designing what your users want

1. **Performance:** Do whatever it takes to make it fast.

2. **Exploration:** Make it easy to try things without fear.

3. **Simplicity:** Make it easy to figure out what to do.
Thanks!