Improving the Development Process (and People) with Metrics-Driven Insights

Open Source Summit North America 2018

@IndeedEng  |  opensource.indeedeng.io
Hi, I’m Jack.

Jack Humphrey, VP Engineering
@IndeedEng
@youknowjack  |  http://jackhumphrey.me
We help people get jobs.
60 countries
30 languages
200M unique visitors
20M jobs
And now: Vancouver!

Indeed Announced Today That It Will Acquire Resume.com to Enable Job Seekers to Build and Download Personalized Resumes for Free

Austin, TX (May 25, 2018) — Indeed announced today that it acquired resume.com, an online service for creating personalized resumes. Over 3.5 million job seekers currently use the site, creating more than 40,000
Indeed Engineering: move fast and try things
Don’t bet on a small number of “great ideas”
Bet on exploring lots of ideas as quickly as possible
Exploring lots of ideas as quickly as possible

1. Hire great people
2. Give them ownership and autonomy
3. Develop great tools
We’ve open sourced some of these tools

Imhotep
Data Analytics Platform

github.com/indeedeng/imhotep
Imhotep: data analytics platform

- Enables rapid exploration & analysis of large time-series datasets
- Query language (IQL), web UI, and distributed backend

scalable  efficient  fast

github.com/indeedeng/imhotep
Imhotep: **scalable**, efficient, fast

Up to **9 million queries/month** at Indeed across ~6K datasets

One popular dataset has **19B events over the last year**

- Last 30 days: **100K distinct queries** invoked **700K times** by 1200 users
- ~85% of invocations are by automated users
Imhotep: scalable, efficient, fast

Another popular dataset: 39B events in the last year, 384 fields
- 5.7TB on disk (146 bytes/event)

No need to sample — use all the data
Imhotep: scalable, efficient, fast

Last 90 days, all queries at Indeed (~20 million)
- Median response time 441 milliseconds
- 70% of queries < 1.6 seconds
- Median **uncached** response time 3 seconds
- Median response time, **uncached** 365-day queries: 9.8s
Metrics

Fast, Iterative Experimentation
Fast, Iterative Improvement
Using metrics to **improve processes**
Using metrics to **improve processes**

01 **Measure** everything
Using metrics to **improve** processes

01 **Measure** everything

02 Ask lots of questions in order to **learn**
Using metrics to **improve** processes

01 Measure everything

02 Ask lots of questions in order to **learn**

03 Based on those learnings, try to **improve**
Using metrics to **improve** processes

01 Measure everything

02 Ask lots of questions in order to **learn**

03 Based on those learnings, try to **improve**

04 Then **measure again** to confirm improvement
Using metrics to **improve** PEOPLE, too

01 **Measure** everything

02 Ask lots of questions in order to **learn**

03 Based on those learnings, try to **improve through coaching**

04 Then **measure again** to confirm improvement
Is measuring process and people a good idea?
Yes.*
* proceed with caution

Yes.*
Goodhart’s Law

When a measure becomes a target, it ceases to be a good measure.
“I am not a number.”
Measures aren’t inherently bad… it’s how you use them.
The metrics should serve the team.
The team shouldn’t serve the metrics.
Alright. But how?

01 Measure everything

02 Ask lots of questions in order to learn

03 Based on those learnings, try to improve

04 Then measure again to confirm improvement
Measure everything (@ Indeed: “put it in Imhotep”)

- Everything that happens in our products
- Everything that happens in our process
  - Git commits
  - JIRA issue updates
  - production deploys
  - wiki edits
  - and more...
Example: Issue Updates

- Everything that happens in our products
- Everything that happens in our process
  - Git commits
  - **JIRA issue updates**
  - production deploys
  - wiki edits
  - and more...
Time for questions

01 Measure everything

02 Ask lots of questions in order to learn

03 Based on those learnings, try to improve

04 Then measure again to confirm improvement
Example hypothesis/complaint:
Translation verification takes too long!

- We track translation work in JIRA
- We track JIRA issue updates in Imhotep
- Question: how long is translation verification taking?
How long are translations in “Pending Verification”?

from jiraactions 2017-01-08 2017-04-02
where issuetype = 'Translation' AND
    prevstatus = 'Pending Verification' AND
    status != 'Pending Verification' AND
    project = 'LOREM'

group by time(1d)

select timeinstate/86400 /* days pending */
How long are translations in “Pending Verification”?

from jiraactions 2017-01-08 2017-04-02
where issuetype = 'Translation' AND
    prevstatus = 'Pending Verification' AND
    status != 'Pending Verification' AND
    project = 'LOREM'
group by time(1d)
select timeinstate/86400 /* days pending */
How long are translations in “Pending Verification”?

from jiraactions 2017-01-08 2017-04-02

where issuetype = 'Translation' AND
    prevstatus = 'Pending Verification' AND
    status != 'Pending Verification' AND
    project = 'LOREM'

group by time(1d)

select timeinstate/86400 /* days pending */
How long are translations in “Pending Verification”?  

from jiraactions 2017-01-08 2017-04-02
where issuetype = 'Translation' AND
    prevstatus = 'Pending Verification' AND
    status != 'Pending Verification' AND
    project = 'LOREM'
group by time(1d)
select timeinstate/86400 /* days pending */
How long are translations in “Pending Verification”? 

from jiraactions 2017-01-08 2017-04-02 
where issuetype = 'Translation' AND 
    prevstatus = 'Pending Verification' AND 
    status != 'Pending Verification' AND 
    project = 'LOREM' 
group by time(1d) 
select timeinstate/86400 /* days pending */
Cumulative time in Pending Verification

Mar 22, 2017, 12:00:00 AM
days pending: 232.573
Ask more questions! Be skeptical.

01 Measure everything

02 Ask lots of questions in order to learn

03 Based on those learnings, try to improve

04 Then measure again to confirm improvement
What’s the issue volume over that timeframe?

```
from jiraactions 2017-01-07 2017-04-29
where issuetype='Translation' AND
       prevstatus='Pending Verification' AND
       status != 'Pending Verification' AND
       project = 'LOREM'
group by time(1d)
select distinct(issuekey) /* number of issues */
```
What’s the issue volume over that timeframe?
What’s the **cumulative** issue volume over that timeframe?

![Graph showing cumulative number of issues over time]

- **Legend:**
  - Red line: number of issues

- **Details:**
  - The graph indicates the cumulative number of issues over a timeframe from February to March 2017.
  - The cumulative number of issues reaches 278 on March 23, 2017, at 12:00:00 AM.
Learn and prioritize

- Good measurements + good questions = learning
- We now have measurements that support our hypothesis, so we can prioritize working on improvement.
Try to improve

01 Measure everything

02 Ask lots of questions in order to learn

03 Based on those learnings, try to improve

04 Then measure again to confirm improvement
Translation verification: there is a better way

- Better way: deploy translations separate from code
- Try the new way incrementally on some projects
  - Project “LOREM”: old process
  - Project “IPSUM”: new process
Does new process reduce verification time?

01 Measure everything

02 Ask lots of questions in order to learn

03 Based on those learnings, try to improve

04 Then measure again to confirm improvement
Let’s look at 90th percentile time in “Pending Verification”

```
from jiraactions 2016-09-15 2017-02-28
where issuetype = 'Translation' AND
    prevstatus = 'Pending Verification' AND
    status != 'Pending Verification'
group by project in ('LOREM','IPSUM')
select percentile(timeinstate, 90)
```
The new process does look faster!

<table>
<thead>
<tr>
<th>#</th>
<th>project in ('LOREM','IPSUM')</th>
<th>percentile(timeinstate,90)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LOREM</td>
<td>1,043,964</td>
</tr>
<tr>
<td>2</td>
<td>IPSUM</td>
<td>154,533</td>
</tr>
</tbody>
</table>

Let’s get LOREM using the new process, then we’ll measure again.
Helping people improve with metrics

01 Measure everything

02 Ask lots of questions in order to learn

03 Based on those learnings, try to improve through coaching

04 Then measure again to confirm improvement
Helping people improve with metrics

HINDSIGHT quarterly eng stats
<table>
<thead>
<tr>
<th>Issues</th>
<th>2016 Q1</th>
<th>2016 Q2</th>
<th>2016 Q3</th>
<th>2016 Q4</th>
<th>2017 Q1</th>
<th>2017 Q2</th>
</tr>
</thead>
<tbody>
<tr>
<td>resolved</td>
<td>103</td>
<td>85</td>
<td>106</td>
<td>51</td>
<td>91</td>
<td>58</td>
</tr>
<tr>
<td>reported</td>
<td>110</td>
<td>93</td>
<td>117</td>
<td>55</td>
<td>85</td>
<td>63</td>
</tr>
<tr>
<td>commented</td>
<td>129</td>
<td>114</td>
<td>110</td>
<td>67</td>
<td>107</td>
<td>76</td>
</tr>
<tr>
<td>reopened</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>deploys</td>
<td>58</td>
<td>29</td>
<td>30</td>
<td>27</td>
<td>28</td>
<td>24</td>
</tr>
<tr>
<td>protests</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>

Fisheye 🎥
459 / 99  332 / 80  303 / 99  191 / 49  349 / 87  219 / 60

Wiki Edits
22     27     55     4      90     21

Review Comments
<table>
<thead>
<tr>
<th>2016 Q1</th>
<th>2016 Q2</th>
<th>2016 Q3</th>
<th>2016 Q4</th>
<th>2017 Q1</th>
<th>2017 Q2</th>
</tr>
</thead>
</table>

Show Graph
Fake Name (Eng - Tech - Eng - Tech)

Issues
resolved
reported
commented
reopened
deploys
protests

Lorem (31)
Ipsum (8)
Dolor (7)
Sit (7)
Amet (4)
Consectetur (4)
Adipiscing (2)
Elit (2)

View All Issues in JIRA
Run JIRA query
What about Number 6 Principle and Goodhart’s Law?

1. We use hindsight as a starting point for discussion.
2. We constantly remind ourselves: don’t treat as targets.
Example conversation: how’s your quality?

Resolved: 100, Reopened: 30
Example conversation: how’s your quality?

Resolved: 100, Reopened: 30

- “Productive, but attempting to ship a lot of buggy code”
Example conversation: how’s your quality?

Resolved: 100, Reopened: 30
- “Productive, but attempting to ship a lot of buggy code”

Be skeptical. Dig into data.
Example conversation: how’s your quality?

Resolved: 100, Reopened: 30

- “Productive, but attempting to ship a lot of buggy code”

Be skeptical. Dig into data.

- Only 10 actual bugs
Example conversation: how’s your quality?

Resolved: 100, Reopened: 30
- “Productive, but attempting to ship a lot of buggy code”

Be skeptical. Dig into data.
- Only 10 actual bugs

Conversation can produce new ideas for individual and team
Measure, question, learn, improve. It can work for process and people.
How do we look at JIRA in Imhotep?
Issues with History

JIRA

Questions

Imhotep Dataset

Actions through Time
Builder JIRA

JIRA Issues
with History

JIRA REST API

Builder

Actions
through Time

TSV
Upload

Imhotep
Dataset

Intrinseo
January 1, 2016 to August 23, 2018

12.3 million actions

1.1M creates, 8.3M updates, 2.9M comments
January 1, 2016 to August 23, 2018

3.2 million actions

200K creates, 1.7M updates, 1.3M comments

262MB on disk: 87 bytes/action
What can we ask Imhotep about Apache projects?

Who reported the most bugs in ASF projects?
Which projects have the most bugs reported?
How are different projects using JIRA differently?
How many people submitted patches, by project?
How much time passes before patches are applied or rejected?
And on and on and on and on…

go.indeed.com/apachejira
Who reported the most bugs in ASF projects? (April - June)

http://imhotep.indeed.tech/iql/q/GXNK8G
Which projects have the most bugs reported? (April - June)

http://imhotep.indeed.tech/iql/q/K88NMK
How are different projects using JIRA differently?

- How many distinct status values in the most active projects?
  http://imhotep.indeed.tech/iql/q/DW2M83

- Compare the statuses used by Apache Beam and Apache Hive:
  http://imhotep.indeed.tech/iql/q/E24GHD
How many unique contributors in 2018?
(projects using Patch Available)

http://imhotep.indeed.tech/iql/q/KW6P27
Who are the contributors to Apache Hive in 2018?

http://imhotep.indeed.tech/iql/q/4CHTKP
How long to get a patch accepted?
(20 most active projects)

http://imhotep.indeed.tech/iql/q/7D6AMD
Why so long, Apache Hadoop MapReduce?

http://imhotep.indeed.tech/iql/q/YYCRP6
How long (median/90th) to get a patch accepted?
(20 most active projects)
Median: http://imhotep.indeed.tech/iql/q/Y48REK
90th Percentile: http://imhotep.indeed.tech/iql/q/H6FP74
go.indeed.com/apachejira

Share what you learn about Apache projects:
Tweet @IndeedEng with #imhotep #apache
Measure, question, learn, improve.
Questions?
@youknowjack  |  jackhumphrey.me

@IndeedEng  |  opensource.indeedeng.io

Share what you learn about Apache projects:
Tweet @IndeedEng with #imhotep #apache