Improving the Development Process with Metrics-Driven Insights

oscon.indeed.tech   |   @IndeedEng   |   github.com/indeedeng
Hi, I’m Jack.
Jack Humphrey, VP Engineering

go.indeed.com/jack | @youknowjack
We help people get jobs.
60 countries

30 languages

200M unique visitors

20M jobs
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 two of these tools

Proctor
A/B Testing Framework

Imhotep
Data Analytics Platform

http://oscon.indeed.tech
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 **5 million queries/week** at Indeed across ~**1500 datasets**

Most popular dataset has **39B events over the last year**

- Last 30 days: **25K distinct queries** invoked **2.2M times** by over **1200 users**
- ~95% of invocations are by a few automated users
Imhotep: scalable, efficient, fast

Most popular dataset: 39B events, 384 fields
- 5.7TB on disk (146 bytes/event)

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

Most popular dataset: queried 2.2M times in last 30 days
- Median response time 159 milliseconds
- 95% of queries < 2 seconds
- Median uncached response time 5.7 seconds
- Median response time, uncached 365-day queries: 37s
These tools enable our approach to development.
We can use this approach to **improve our processes**

01 **Measure** everything

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

03 Based on those learnings, make changes (**deliver**)

04 Then **measure** to confirm improvement
We can use this approach to help people improve

01 Measure everything

02 Ask lots of questions in order to learn

03 Based on those learnings, make changes (deliver)

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

* proceed with caution
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.
Process improvement: an example

 Measure

 Deliver

 Learn
Measure (a.k.a. “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...
- We translate each string change or new string to 30 languages.
- We use a custom JIRA issue type to track: 1 per language.
- Verification has been historically tedious...
  - So let’s **measure** how long we spend verifying translations.
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 timeinstance/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”? 

```sql
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 */
```
Verification time is adding up

![Graph showing cumulative days pending over time with a peak on March 22, 2017, at 12:00:00 AM, with 232.573 days pending.](https://via.placeholder.com/720x405)
Be skeptical and question all measurements.

- Can we dig into the data to better understand it?
- What other information is needed to interpret?
- What are the sources of noise?
- Do we need to iterate on the measurement itself before it is generally useful?
What’s the issue volume over that timeframe?

```sql
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?
Learn and prioritize

- Good measurements + good questions = learning
- We now have a measurement that supports our hypothesis, and we can prioritize improvement.
Translation verification: there is a better way

- Better way: deploy translations separate from code
- Does new process reduce verification time?
  - Project "LOREM": old process
  - Project "IPSUM": new process
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>1,198,497</td>
</tr>
<tr>
<td></td>
<td>12 days</td>
<td>1,043,964</td>
</tr>
<tr>
<td>2</td>
<td>IPSUM</td>
<td>154,533</td>
</tr>
<tr>
<td></td>
<td>1.8 days</td>
<td>154,533</td>
</tr>
</tbody>
</table>

Worth implementing in LOREM, then we’ll measure again
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>
<tr>
<td>Fisheye</td>
<td>459 / 99</td>
<td>332 / 80</td>
<td>303 / 99</td>
<td>191 / 49</td>
<td>349 / 87</td>
<td>219 / 60</td>
</tr>
<tr>
<td>Wiki Edits</td>
<td>22</td>
<td>27</td>
<td>55</td>
<td>4</td>
<td>90</td>
<td>21</td>
</tr>
<tr>
<td>Review Comments</td>
<td>2016 Q1</td>
<td>2016 Q2</td>
<td>2016 Q3</td>
<td>2016 Q4</td>
<td>2017 Q1</td>
<td>2017 Q2</td>
</tr>
</tbody>
</table>
fakeproject / resolved / 2017 Q2: 8 projects, 65 issues

- LOREM (31)
- IPSUM (8)
- DOLOR (7)
- SIT (7)
- AMET (4)
- CONSECTETUR (4)
- ADIPISCING (2)
- ELIT (2)

View All Issues in JIRA
Guard against 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 my 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, and learn.
It can work for process and people.
Tracking the Issue Tracker
Hi, I’m Kevin.
Kevin Binswanger, Software Engineer
## Known unknowns: **quality**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Reopens</td>
</tr>
<tr>
<td>2</td>
<td>Escaped bugs</td>
</tr>
<tr>
<td>3</td>
<td>Time and Severity</td>
</tr>
<tr>
<td></td>
<td>Known unknowns: <strong>engagement</strong></td>
</tr>
<tr>
<td>---</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>1</td>
<td>How many committers</td>
</tr>
<tr>
<td>2</td>
<td>How much they are committing</td>
</tr>
<tr>
<td>3</td>
<td>Bus Factor</td>
</tr>
</tbody>
</table>
## Known unknowns: velocity

|   |  
|---|---
| 1 | Committed > Reviewed > Verified  
| 2 | Inside the team vs Outside the team  
| 3 | Places we can automate  

January 1, 2016 to May 1, 2017

3.9 million actions
January 1, 2016 to May 1, 2017

1.6 million actions
We Imported All the Activity from the Apache JIRA Instance, You’ll Never Believe What Happened Next!

#4 will surprise you!

http://go.indeed.com/apachejira
http://go.indeed.com/apachejira

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

http://go.indeed.com/apachejira