Measuring and Improving the Efficiency of Software Delivery

Open Source Summit North America 2019

Jack Humphrey, VP Engineering, Indeed
Siva Dosapati, Engineering Director, Indeed
We help people get jobs.
Indeed is the #1 job site worldwide

Source: comScore, Total Visits, March 2018
250M unique monthly visitors

150M resumes

150M ratings and reviews

10 jobs added/second

600M salaries

1 Source: Google Analytics, Unique Visitors, September 2018
FOSS Contributor Fund

go.indeed.com/foss-fund-6m
Efficiency Matters

“Faster is better - even in open source.”
(Nicole Forsgren, PhD; earlier this morning)
Why should you care about the efficiency of software delivery?
Speed of learning and innovation
Developer morale and productivity
Time to market
Unit cost economics
@IndeedEng

> 200% more developers
> 200% more developers
130% more delivery
28% throughput decline
Continuous, incremental improvement
Focus on “good change”
**Measure** and evolve over time
How do you measure the efficiency of software delivery?
Find the delivery metrics that are relevant and useful to you.
Examples of efficiency metrics
Total number of changes that are complete and can be deployed to production over a given time interval.
### Average # story points per sprint

<table>
<thead>
<tr>
<th>Sprint 1</th>
<th>Sprint 2</th>
<th>Sprint 3</th>
<th>Sprint 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>11</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>15</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Velocity:**
- Sprint 1: 38
- Sprint 2: 29
- Sprint 3: 38
- Sprint 4: 39
- Overall: 36
Average time to deliver a unit of work to production (see also: Cycle Time)

**DELIVERY LEAD TIME (DLT)**

In Progress | Pending Review | Pending Merge | Pending Verification | Pending Closure | Closed / In Production
--- | --- | --- | --- | --- | ---
4.0d | 2.0d | 3.2d | 2.8d | .5d |

**DELIVERY LEAD TIME = 12.5 DAYS**
How frequently you deploy code

Source: GoCD, https://go.indeed.com/T72ZF3
Limiting WIP boosts collaboration and increases throughput

Capacity: 95%
Avg Speed: 19 mph
Throughput: 70 cars/segment/hour

Capacity: 60%
Avg Speed: 60 mph
Throughput: 80 cars/segment/hour

Inspiration: https://go.indeed.com/NWZNA8
Examples of quality metrics that impact efficiency
# of major bugs impacting customers

COST

- Reqs
- Design
- Dev
- Test
- Production
What % of deploys result in failure?
MEAN TIME TO RESTORE (MTTR)

How long does it take to restore service after failure?
How did we choose and develop our metrics?
WE CHOSE DELIVERY LEAD TIME

Faster value delivery

Increased developer throughput

Eliminate Bottlenecks

Easily Measurable
Exploring Jira data
Exploring Jira data with Imhotep

github.com/indeedeng/imhotep
Issues with History

JIRA
REST API

Builder

TSV
Upload

Actions through Time

Imhotep Dataset

github.com/indeedeng/imhotep-builder-jira
FROM jiraactions 366d 1d
WHERE issuetype IN ("New Feature", "Improvement", "Bug")
    prevstatus in ("In Progress", "Pending Review", ...)
    fieldschangedtok="status" status != prevstatus
SELECT AVG_OVER(issuekey, timeinstate)/60/60/24 AS DLT_Days
DELIVER INSIGHTS

VELOCITY INSIGHTS DASHBOARD

Peregrine
A Velocity Insights product
DELIVER
INSIGHTS

MONTHLY EMAILS

Your velocity report for June

Blazar Team: [Redacted]

Jul 5, 2019, 4:03 PM

to me

indeed peregrine

Your Teams

Velocity Report
June 2019

Here's your overall velocity snapshot:

2104 tickets closed

<table>
<thead>
<tr>
<th>DLT (Mean)</th>
<th>DLT (Median)</th>
<th>Major prod. esc.</th>
<th>Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>11.5 days</td>
<td>6.4 days</td>
<td>40 bugs</td>
<td>983 deploys</td>
</tr>
</tbody>
</table>

37.34% decrease
2.22% increase
0.09% decrease
1.21% decrease

Want to know how we generated these numbers?
DELIVER INSIGHTS

STATUS UPDATE WIDGETS

**4.28**
Avg Delivery Lead Time: Days per Issue (12w)
05-07-2019 to 05-20-2019
Past Two Weeks by Day

**84.0**
Deploys per Project (12w)
05-07-2019 to 05-20-2019
Past Two Weeks by Day
Did we face any challenges?
Feeling some skepticism?
Won’t people just game the stats?
Goodhart’s Law

When a measure becomes a target, it ceases to be a good measure.
Sacrificing values: agility, quality, reliability
Losing sight of continuous improvement
Empower your teams to:

Forget the metric, live the values (e.g. agility, quality, reliability)

Evolve as needed -- hypothesize, test new ideas, get comfortable with change
Identify metrics that will show undesirable outcomes

Watch those metrics and investigate

Examples for DLT: throughput, bugfix delivery rate
Won’t people just game the stats?

How will I be evaluated on this metric?
Don’t evaluate people using only metrics

Celebrate real wins for the team’s ability to deliver
Won’t people just game the stats?
How will I be evaluated on this metric?
This metric is sometimes wrong!
BUT IS IT USEFUL?

Will it encourage the right action?
Is it meaningful in aggregate …across teams? …over time?
It doesn’t have to be perfect.
Make it clear how you’re using it.
And iterate on the measurement.
Issue sizing and estimation
Outliers and averages
Change management and culture
How have we rolled out this focus?
APPROACH

- DEFINE
- COMMUNICATE
- ALIGN
- DEVELOP
- TRAIN
- INCENTIVIZE
- MEASURE
Reduce avg DLT by 50% in 2019

14 days  7 days

2018    2019
Communication from senior leadership
Newsletters, videos, emails, company updates, Q&A
Repeated on a regular cadence
Every group/team set a DLT goal

**Delivery Lead Time by Group, Q1 Vs Q2**
Tools to view and explore DLT

![Delivery Lead Time Chart](image-url)
Efficiency improvement workshops
Value individual contributions to efficiency improvement

Modified performance/career rubrics to clarify this value
REPORT ON PROGRESS

Dashboards
Slack notifications
Content widgets
Leadership reviews
Company updates
What should you do with all this?
What does it mean for your domain?
What does it mean for your team(s)?
How do your teams feel about it?
How will you measure it?
Focus on continuous improvement
Develop metrics & iterate on them
Monitor for undesirable outcomes
Celebrate all the wins
Questions?

indeed® engineering
indeed.tech
blog, talks, open source, and great ideas
What’s next for Indeed?
Do we see the expected correlations?
Do we avoid the unwanted behaviors?
Have we maintained sufficient quality?
EFFICIENCY OF VALUE DELIVERY

Experiment Lead Time
Experiment Throughput
Experiment Yield