Measuring Continuous Delivery

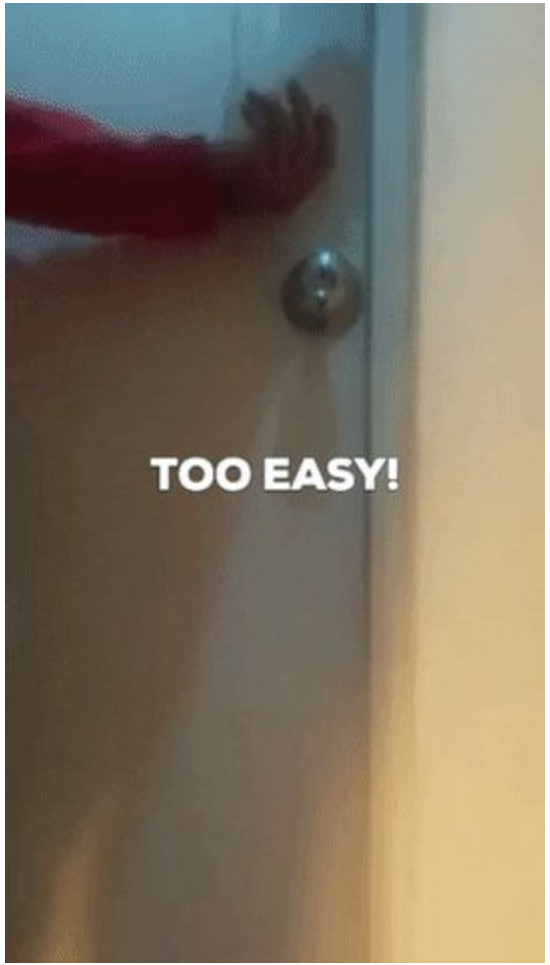
TL;DR:

Measure the system, not teams or individuals

Don’t turn metrics into targets

Measure time and quality

Wouter Lagerweij | @wouterla | wouter@lagerweij.com


TOO EASY!


ABUS
Systemeinstellungen
wAppLoxx
Zeitpläne
Zutrittsgruppen
Benutzer
Zuordnungsansicht
Ereignisse
Benutzer
Matthias Schmidt
Thomas Huber
Eva Schneider
Manuel Becker
Alexander Fischer
Eingangstür
Abbrechen
Offen
wAppLoxx Control
DESIGN AWARD 2017
PLUS X AWARD
ausgezeichnet als:
BESTES PRODUKT 2016/2017
PLUS X AWARD
2016 ausgezeichnet für:
Innovation
High Quality
Bedienkomfort
Funktionalität

ADDITIONAL INFORMATION

Updated
February 9, 2018

Size
13M

Installs
1,000+

Current Version
1.3

Requires Android
5.0 and up

Content Rating
PEGI 3
Learn More

Permissions
View details

Report
Flag as inappropriate

Offered By
Google Commerce Ltd

Developer
Visit website
f.lasch@abus-sc.com
Privacy Policy
ABUS Security-Center
Linker Kreuthweg 5
86444 Affing Germany


EUROSPEC
MP
10
MULTI-PIN
SECURITY

Wouter Lagerweij

@wouterla

wouter@lagerweij.com

How often do you deploy?

< once every 6 months	1 per month - 1 per 6 months	1 per week - 1 per month	1 per day - 1 per week	1 per hour - 1 per day	On demand, multiple a day

How long does it take for code to get to production?

Average time between when a developer checks code into source control, and it running on production

> 6 months	1 month - 6 months	1 week - 1 month	1 day - 1 week	< 1 day	< 1 hour

How often does a deploy fail?

Needs a rollback, needs a (hot)fix

> 75 %	50 - 75 %	25 - 50 %	15 - 25 %	> 15 %

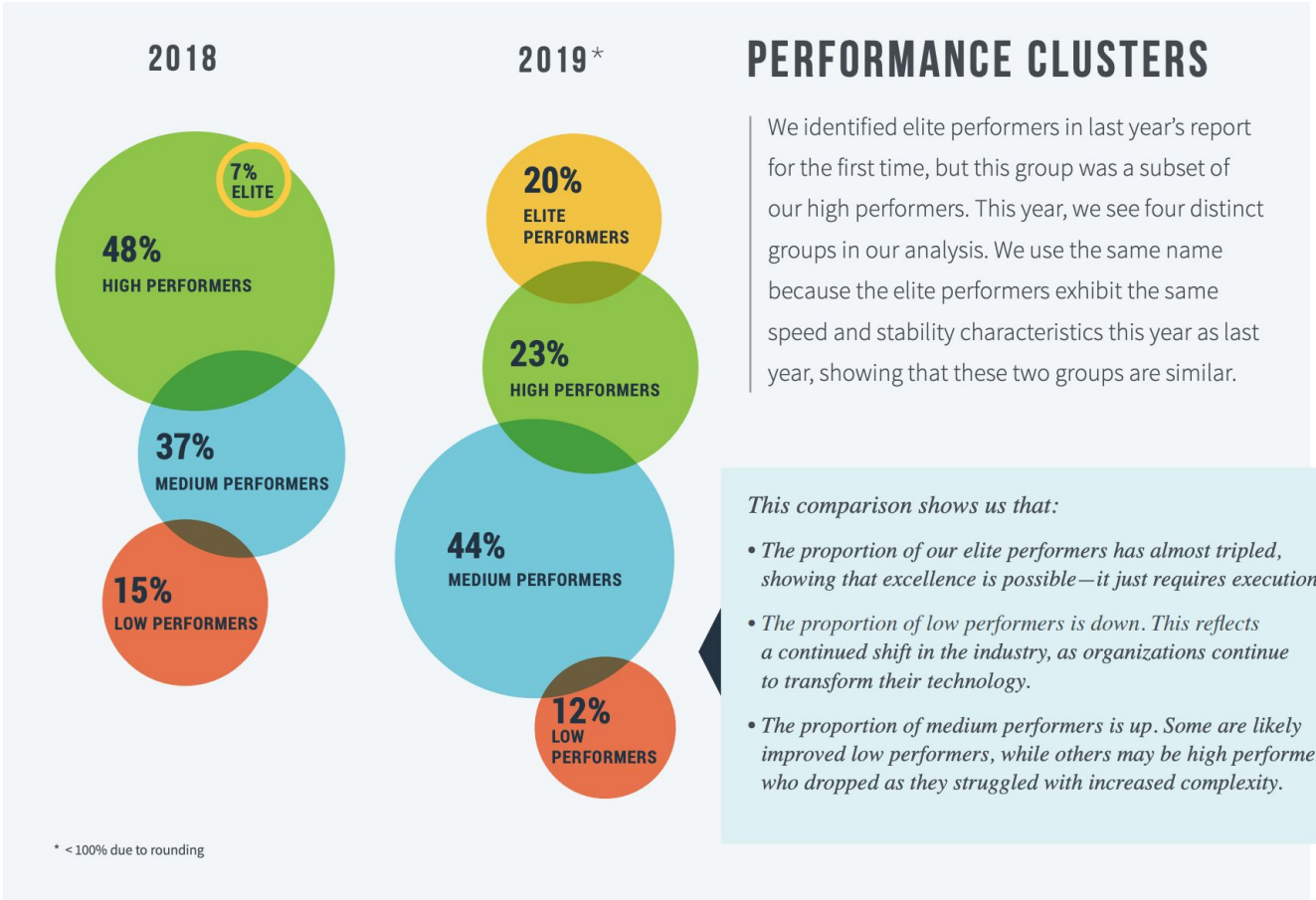
How long to fix an issue?

MTTR

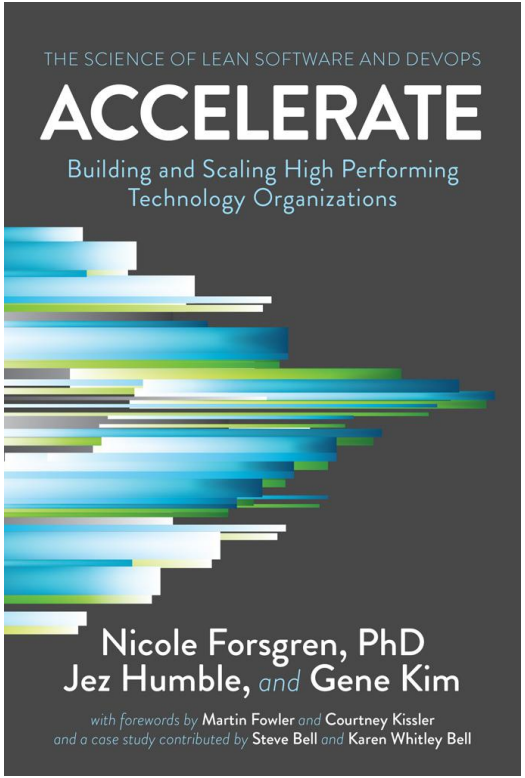
> 1 month	1 week - 1 month	1 day - 1 week	< 1 day	< 1 hour

Aspect of Software Delivery Performance*	Elite	High	Medium	Low
Deployment frequency For the primary application or service you work on, how often does your organization deploy code to production or release it to end users?	On-demand (multiple deploys per day)	Between once per day and once per week	Between once per week and once per month	Between once per month and once every six months
Lead time for changes For the primary application or service you work on, what is your lead time for changes (i.e., how long does it take to go from code committed to code successfully running in production)?	Less than one day	Between one day and one week	Between one week and one month	Between one month and six months
Time to restore service For the primary application or service you work on, how long does it generally take to restore service when a service incident or a defect that impacts users occurs (e.g., unplanned outage or service impairment)?	Less than one hour	Less than one day[a]	Less than one day[a]	Between one week and one month
Change failure rate For the primary application or service you work on, what percentage of changes to production or released to users result in degraded service (e.g., lead to service impairment or service outage) and subsequently require remediation (e.g., require a hotfix, rollback, fix forward, patch)?	0-15%[b,c]	0-15%[b,d]	0-15%[c,d]	46-60%

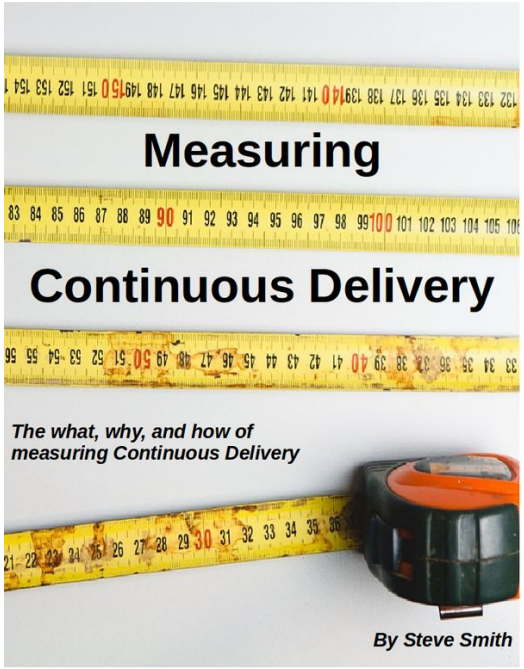
Source: The State of DevOps Report 2019



Source: [The State of DevOps Report 2019]



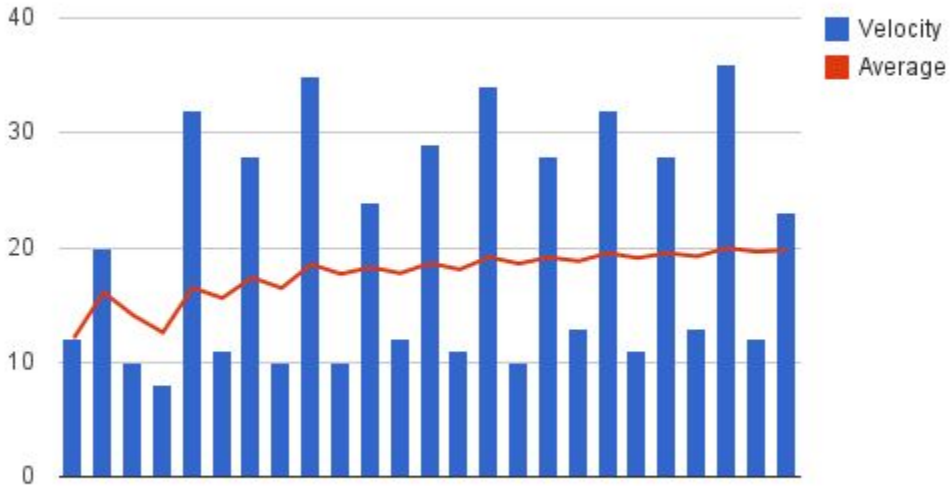
[Accelerate on Amazon]



https://leanpub.com/measuringcontinuousdelivery

Why those?

What about Velocity?



“Managers who can’t measure what they value
will instead
value what they can measure”

-- Russell Ackoff

Goodhart's Law

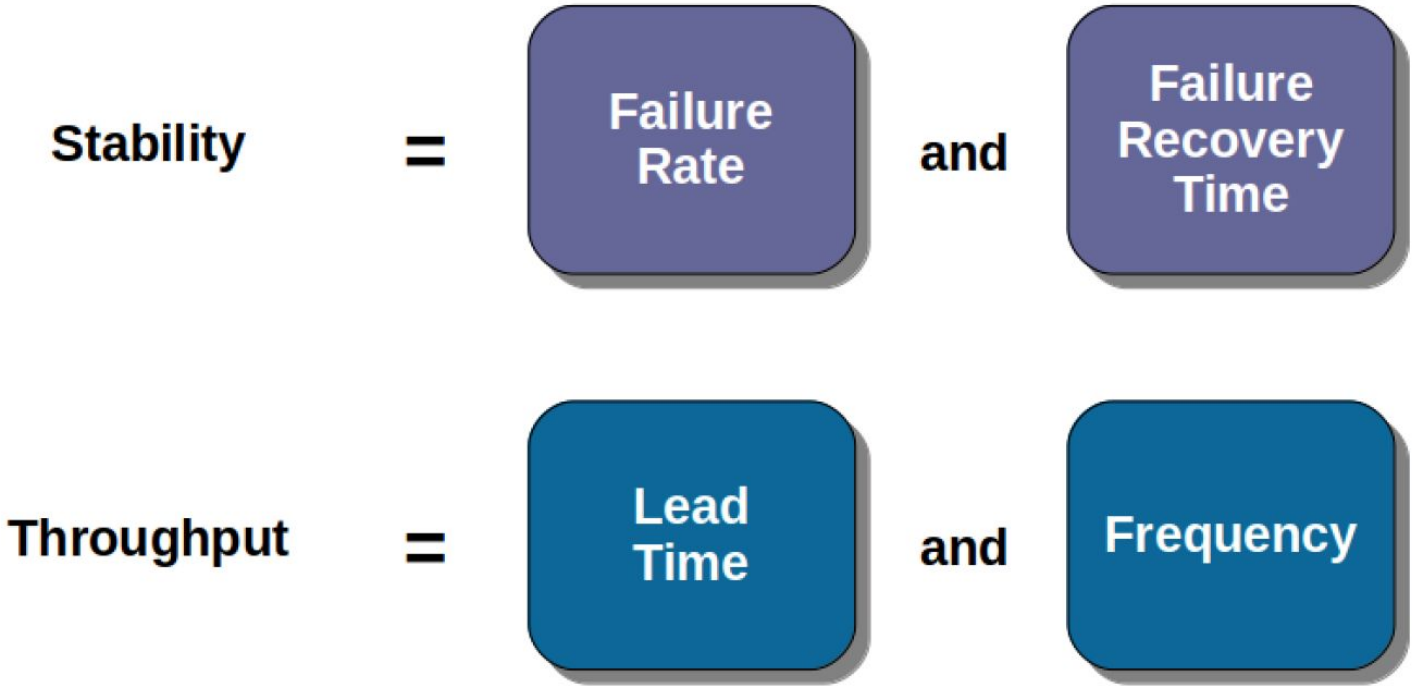
**"When a measure becomes a target,
It ceases to be a good measure"**

https://en.wikipedia.org/wiki/Goodhart's_law

What are you measuring that has side-effects?

What about utilization?

Measure the system, not the people



Stability and throughput measures

Source: Measuring Continuous Delivery - Steve Smith

A single number is meaningless.
Combine metrics, and watch trends.

Throughput

- Deployment frequency

  “Twice a year”

- Lead time for changes
- Mean time to recover
- Change failure rate

Throughput

- Deployment frequency

  “Twice a year”

- Lead time for changes

  “Three months”

- Mean time to recover
- Change failure rate

Throughput

- Deployment frequency

  “Twice a year”

- Lead time for changes

  “One day”

- Mean time to recover
- Change failure rate

A change in a metric is an invitation for a conversation

Less focus on accountability, more focus on giving people a chance to give their account

-- David J. Bland / Tom Looy

Measuring Continuous Delivery

TL;DR:

Measure the system, not teams or individuals

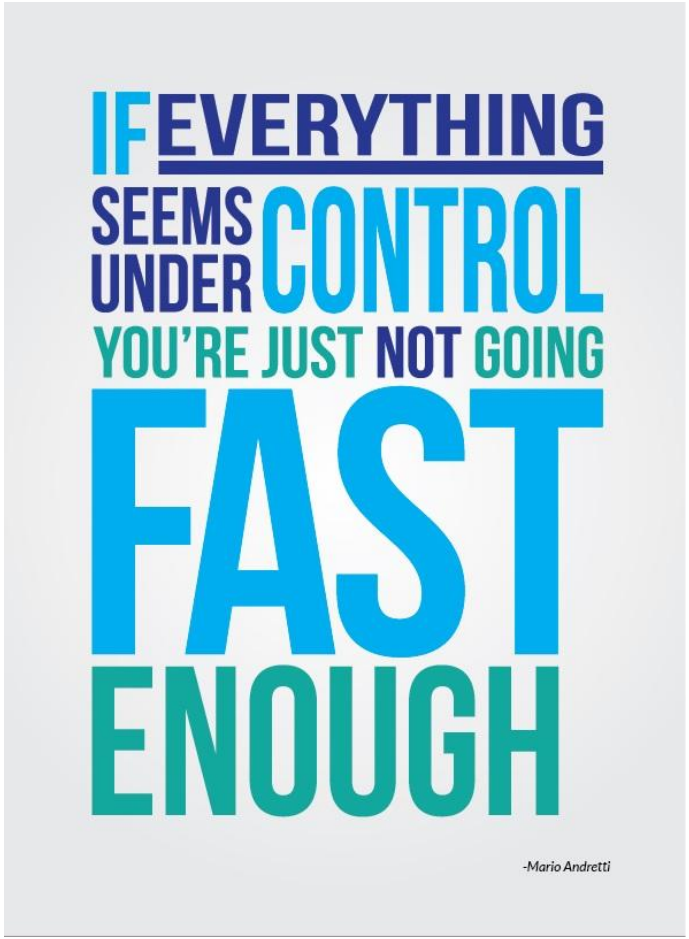
Don’t turn metrics into targets

Measure time and quality

Wouter Lagerweij | @wouterla | wouter@lagerweij.com

No metric without a goal


Do you want to be predictable?
FROZEN
Royal Reception
wait time
120
minutes from
this point
Or fast?


IF EVERYTHING
SEEMS UNDER CONTROL
YOU'RE JUST NOT GOING
FAST
ENOUGH
-Mario Andretti

Throughput and stability

- Deployment frequency

  “How often do we deploy to production?”

- Lead time for changes

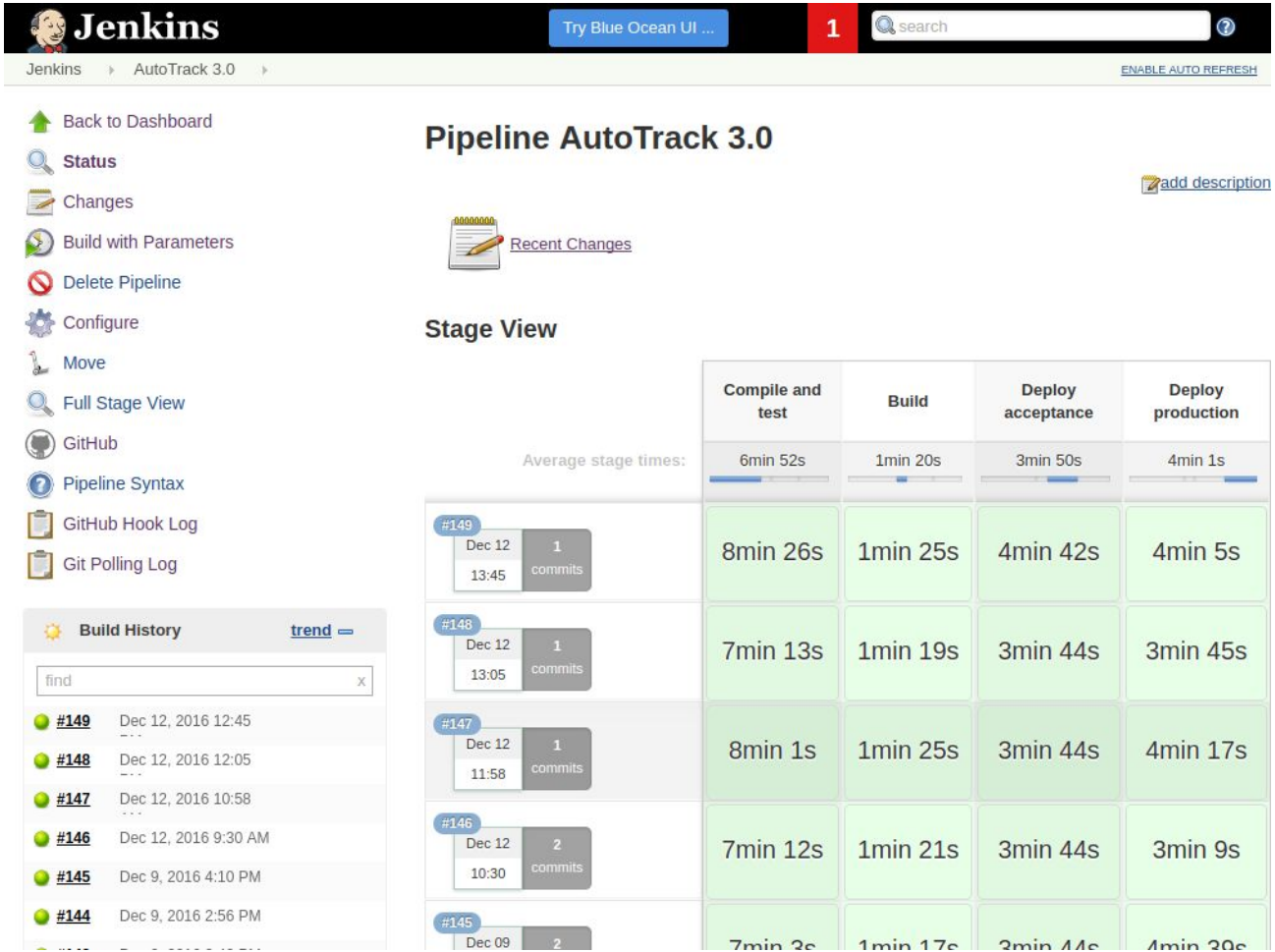
  “How much time from developer code check-in to running in production?”

- Mean time to recover

  “How long between detection of production issue and resolving it”

- Change failure rate

  “How often does a deployment fail?”


Jenkins
Try Blue Ocean UI ...
1
search
Jenkins
AutoTrack 3.0
ENABLE AUTO REFRESH
Back to Dashboard
Status
Changes
Build with Parameters
Delete Pipeline
Configure
Move
Full Stage View
GitHub
Pipeline Syntax
GitHub Hook Log
Git Polling Log
Build History
trend
find
#149 Dec 12, 2016 12:45
#148 Dec 12, 2016 12:05
#147 Dec 12, 2016 10:58
#146 Dec 12, 2016 9:30 AM
#145 Dec 9, 2016 4:10 PM
#144 Dec 9, 2016 2:56 PM
Pipeline AutoTrack 3.0
add description
Recent Changes
Stage View
Compile and test
Build
Deploy acceptance
Deploy production
Average stage times:
6min 52s
1min 20s
3min 50s
4min 1s
#149 Dec 12 13:45 1 commits
8min 26s
1min 25s
4min 42s
4min 5s
#148 Dec 12 13:05 1 commits
7min 13s
1min 19s
3min 44s
3min 45s
#147 Dec 12 11:58 1 commits
8min 1s
1min 25s
3min 44s
4min 17s
#146 Dec 12 10:30 2 commits
7min 12s
1min 21s
3min 44s
3min 9s
#145 Dec 09 2

‘Product Teams’: Value

See progress from business perspective
Redirect teams when needed

Team
Skills Shift

Deliver Value
★★

Ship on market cadence
Capture value frequently
Reveal obstructions early

Organizational
Structure Shift

Optimize Value
★★★

Make excellent product decisions
Eliminate handoffs
Speed decision making

⭐⭐⭐ ‘Product Teams’: Value

“Value is what you like”

-- Ron Jeffries

http://ronjeffries.com/xprog/articles/value-is-what-you-like/

‘Product Teams’: Value

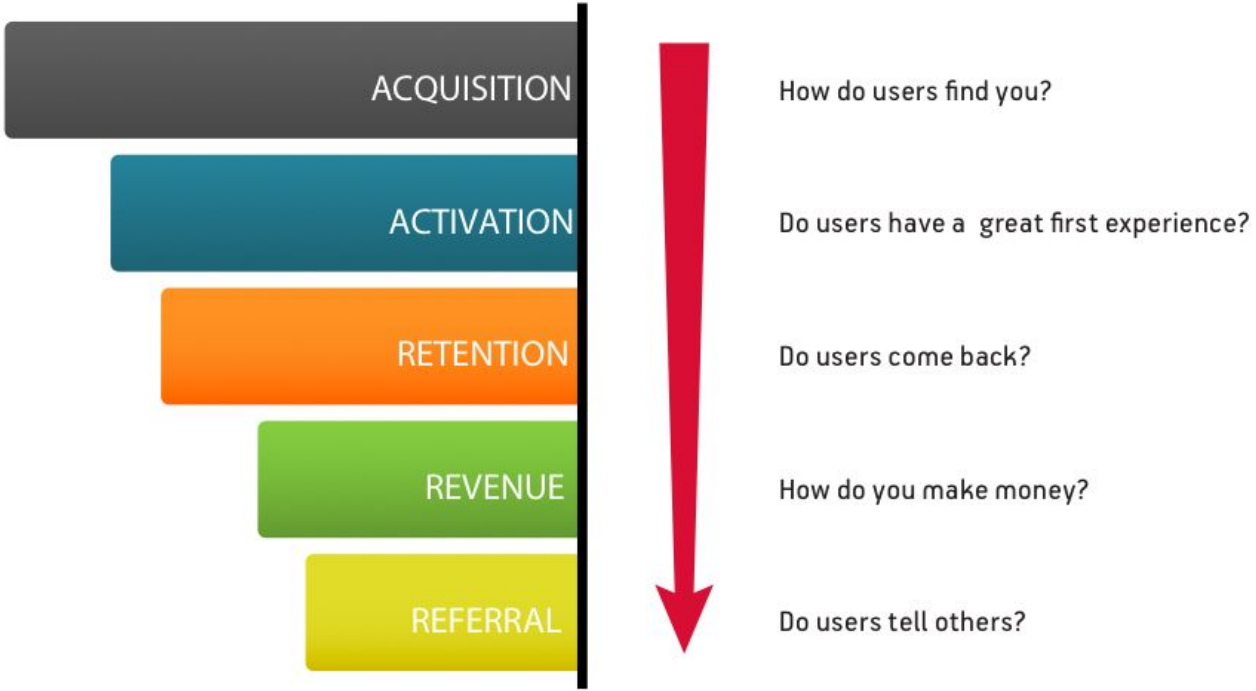
“What is your business model?”

-- Me, repeatedly

‘Product Teams’: Value

- Web: Pirate metrics!
- Internal applications: UX analytics

‘Product Teams’: Value



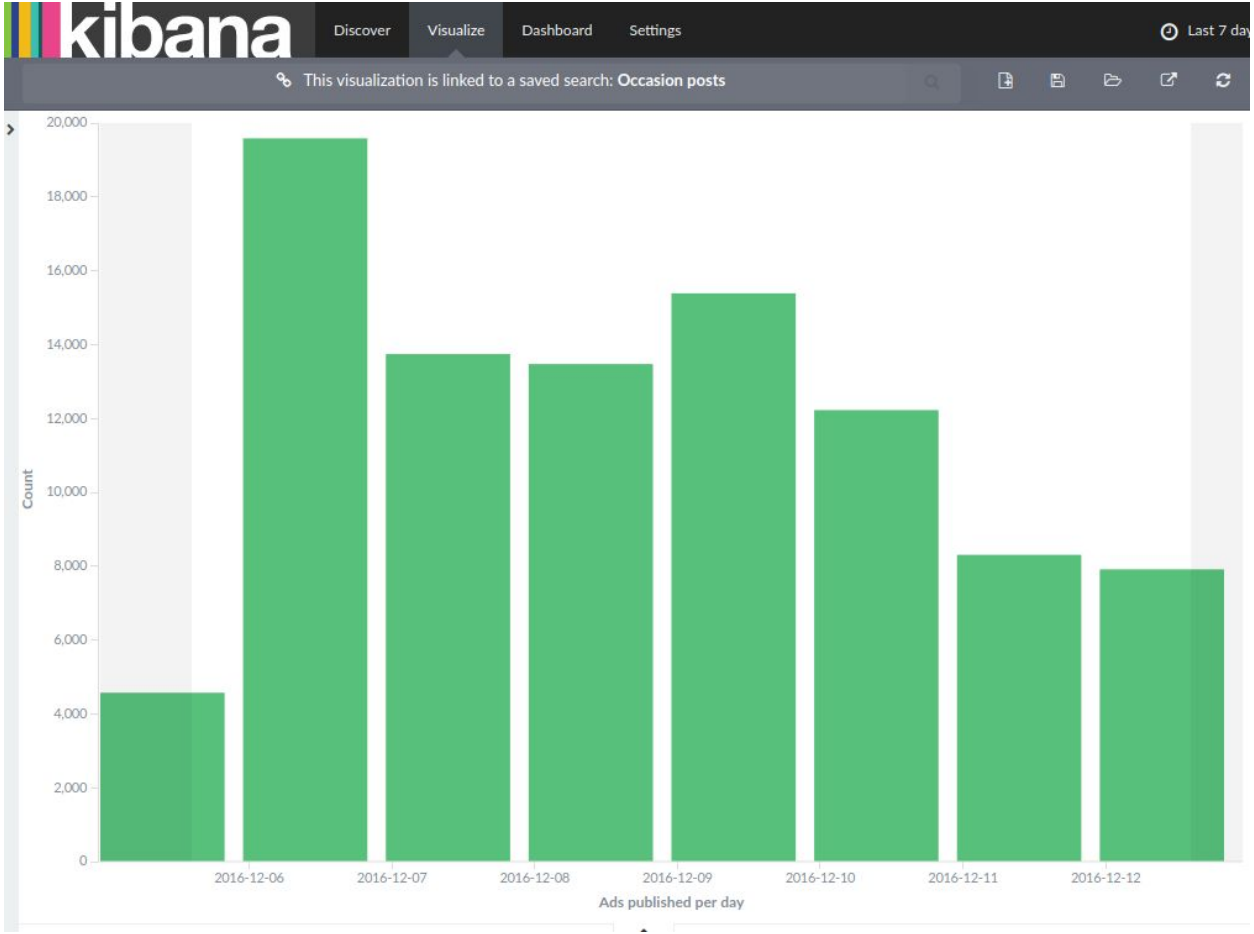
http://500hats.typepad.com/500blogs/2007/06/internet-market.html | The Startup Owner’s Manual

‘Product Teams’: Value

Example Conversion Metrics

Category	User Status	Conv %	Est. Value
Acquisition	**Visit Site** (or landing page, or external widget)	100%	$.01
Acquisition	**Doesn't Abandon** (views 2+ pages, stays 10+ sec, 2+ clicks)	70%	$.05
Activation	**Happy 1st Visit** (views X pages, stays Y sec, Z clicks)	30%	$.25
Activation	**Email/Blog/RSS/Widget Signup** (anything that could lead to repeat visit)	5%	$1
Activation	**Acct Signup** (includes profile data)	2%	$3
Retention	**Email Open / RSS view -> Clickthru**	3%	$2
Retention	**Repeat Visitor** (3+ visits in first 30 days)	2%	$5
Referral	**Refer 1+ users who visit site**	2%	$3
Referral	**Refer 1+ users who activate**	1%	$10
Revenue	**User generates minimum revenue**	2%	$5
Revenue	**User generates break-even revenue**	1%	$25

http://startitup.co/guides/374/aarrr-startup-metrics


kibana
Discover
Visualize
Dashboard
Settings
Last 7 day
This visualization is linked to a saved search: Occasion posts
20,000
18,000
16,000
14,000
12,000
10,000
8,000
6,000
4,000
2,000
0
Count
2016-12-06
2016-12-07
2016-12-08
2016-12-09
2016-12-10
2016-12-11
2016-12-12
Ads published per day

Actionable Metrics

Work in Progress

Metric: WIP

Distance: Direct

Scale: **Individual**, Team, Department, Organisation

Actionable: Yes

Feedback Cycle: From minutes to years

Individual measurements:

- How many tasks is any person working on?
- How many features? Projects?

Practices:

- Personal kanban, Getting Things Done
- King/Servant pattern
- TDD
- Story slicing

Stop Starting, Start Finishing

Goal: Focus

Metric: WIP

Distance: Direct

Scale: Individual, Team, Department, Organisation

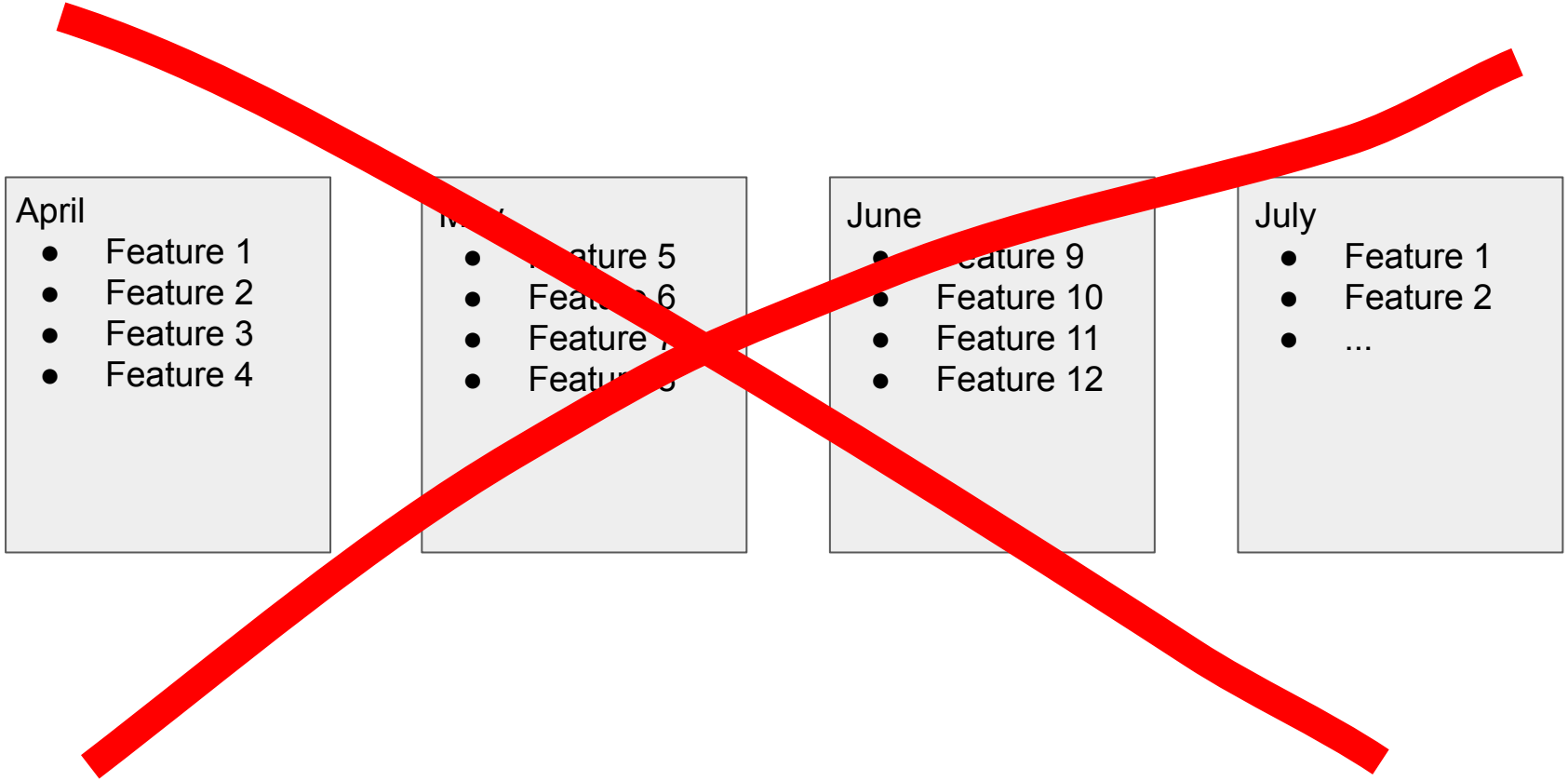
Actionable: Yes

Feedback Cycle: From minutes to years

Style Starting,
Start Finishing

⭐⭐ ‘Continuous Delivery’: Capability

- Deployment frequency
- Lead time for changes
- Mean time to recover
- Change failure rate


April
Feature 1
Feature 2
Feature 3
Feature 4
June
Feature 10
Feature 11
Feature 12
July
Feature 1
Feature 2
...

Goals

Goal:
Improve product detail views coming via search

Goal:
Improve retention active users first 30d

Goal:
Improve conversion product detail > cart

Features

Goal:
Improve product detail views coming via search

Idea:
Improve relevancy results

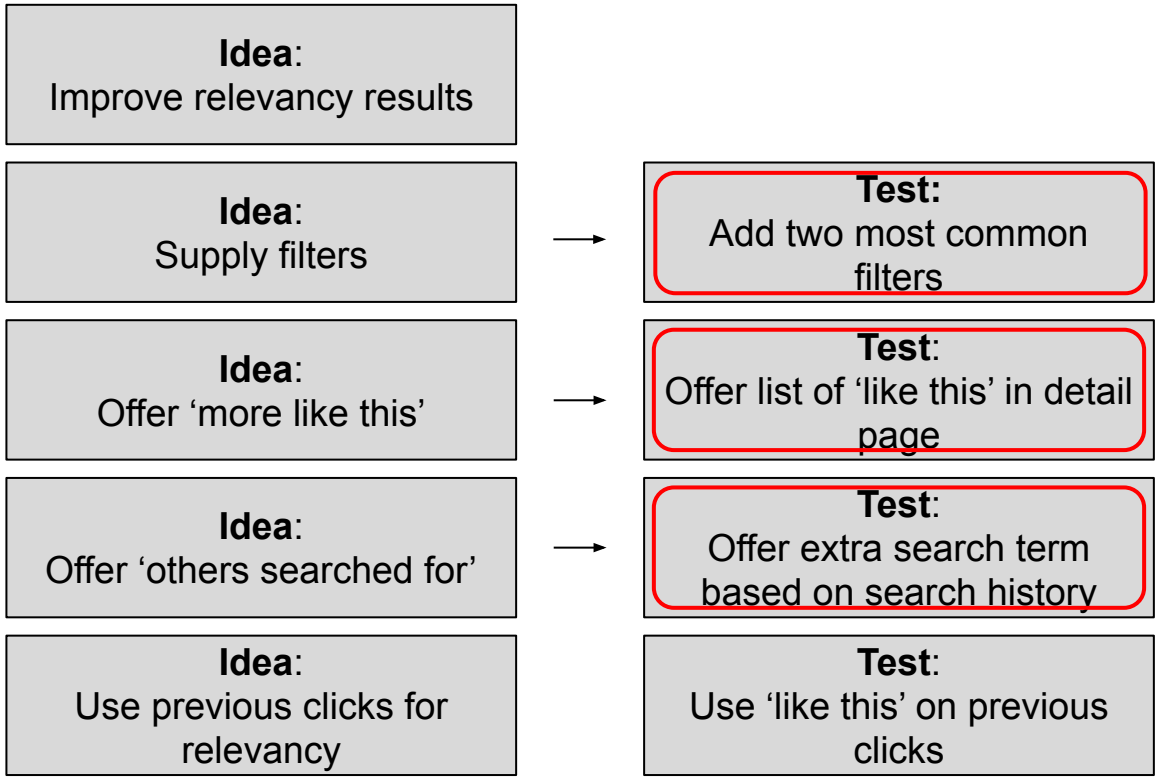
Idea:
Supply filters

Idea:
Offer ‘more like this’

Idea:
Offer ‘others searched for’

Idea:
Use previous clicks for relevancy

Tests


Idea:
Improve relevancy results
Idea:
Supply filters
Test:
Add two most common filters
Idea:
Offer ‘more like this’
Test:
Offer list of ‘like this’ in detail page
Idea:
Offer ‘others searched for’
Test:
Offer extra search term based on search history
Idea:
Use previous clicks for relevancy
Test:
Use ‘like this’ on previous clicks

Results

Test: Add two most common filters	**1.5% more searches** **1% more detail views**
Test: Offer list of ‘like this’ in detail page	**.5% more searches** **4% more detail views**
Test: Offer extra search term based on search history	**5% more searches** **2% more detail views**