Metrics Driven Development and DevOps

- Ernest Mueller (@ernestmueller)
- Karthik Gaekwad (@iteration1)
Karthik Gaekwad

- Senior Engineer @Signal Sciences
- Previous:
  - 10 years building products-agile/cloud/devops teams
Ernest Mueller

- Product Manager at Copperegg
- Previous:
  - 20 years in IT – dev, ops, management
  - @ernestmueller

@iteration1 #Agile2014 https://twitter.com/ernestmueller
Our Goal For You Today

• Empower you with new ideas to bring your organization together!
• Metrics. What are they?
• How to use metrics for good, as illustrated by three Epic Rap Battles of History!
  – Dev vs Ops (What is this... DevOps?)
  – Small vs Large Org
  – Scrum vs Kanban

@ernestmueller #Agile2014 @iteration1
Audience Survey
What Are Metrics?

• A quantifiable measure of any component or process whose change is of interest to your business.
  – Business!
  – Application!
  – System!
  – People!
  – Process!
  – Not: Meaningless numbers!

@ernestmueller #Agile2014 @iteration1
Round One

Dev vs Ops
Help the Business
Build THE product
That you can’t live without...

@ernestmueller  #Agile2014  @iteration1
Teamwork
Agile Teams are faced with tough decisions
“If you don’t know where you’re going, any road will take you there.”
Metrics help with this problem.
Story Time: Our context

• Tasked with building new cloud business for the organization.
• Understand how cloud technologies can impact bottom line.
• Build products customers will want from the new business unit.
  – Read, startup inside a bigger organization
Lean Startup Applied

• Used ‘Lean Startup’ ideas to power new area.
  – Able to define an MVP (Minimum Viable Product).
  – Easier to define workflow for something brand new.
  – No confusion with existing processes.

• Once we started to see value, retrofitted to other parts of the org.
Showing progress

• Initially – we had weekly progress/status meetings with stakeholders.

• Cross functional team with business/marketing/engineering.
Soon this happened..
Talking == Not actionable
Metrics

• Pivot: change conversations to metrics instead.
• Agreed on metrics that we wanted to track
  – Stakeholder input

• “What do you want out of this?”
• “How quickly do you want this?”

• …Okay, let’s measure this!
Tracked Metrics

• Tracked actionable metrics (dev and business):
  – # Users signing up per week
  – # Active sessions per day/week
  – # of compiles sent per week
  – # unique data points sent per week
Pro Tip: Metrics

• Link all your metrics from one dashboard.
  – Business (Ex: User logins)
  – Dev (Ex: Performance metrics)
  – Ops (Ex: DB CPU Usage)
• One bookmark to rule them all.
Pro Tip: Metrics

• Don’t use yet another username/password scheme.
• You’ll lose your users really fast!
Pro Tip: Metrics

• Try to use a tool that can handle different kinds of metrics.
• Shoutouts:
  – Statsd
  – Datadog
End Result

- Business and engineering on the same page.
- Management looking at metrics without having “meetings to look at metrics”.
- Became a part of the culture.
- Innovate faster because different teams were in sync.
Operations

What is it?

Why Do You Care?
Other Kinds Of Operations

• Wikipedia quothe:
• Business operations is the harvesting of value from assets owned by a business
• Operations management is {...} overseeing, designing, and controlling the process of production and redesigning business operations in the production of goods or services.
Technical Operations


• Without the ability to
  – Release changes
  – Quickly respond to change
  – Provide a service without interruption
  – Operate cost effectively

Your service is borked.
What Does Operations Do?

• Build Servers, OS, Virtualization/Cloud
• Install/Upgrade Software
• Install Applications/Release Process/Move to Prod
• Configure Network, Load Balancers, Storage, etc.
• Security testing, reporting, and hardening
• Reliability (scaling, backups)
• Performance management (apps, systems)
• Scalability (capacity planning to autoscaling)
What Else Does Operations Do?

- Availability – Responsible for service being up
- Incident Response
- Fulfill Requests
- Budgeting/Contracts/Cost Tracking/Reduction
- Monitor all of that
- Much more
- So besides “they run the services,” the critical final piece of your value chain, they have access to many of the things you want metrics from
Code – Operations =
Story Time: Black Friday

• Every year, a huge spike in usage
• Uptime and performance critical to retailers during the period
• Product directly contributed to conversion
• Metrics crucial to plan the period, execute through the period, report how we did
Metrics From...

From:
- Servers
- Applications/App Logs
- App Servers/Software
- Network
- Data Stores
- Web Servers/CDNs
- Client Browsers
- Alerts
- Tickets

Using:
- Open source monitoring tools (Zabbix, nagios)
- SaaS (SumoLogic, PagerDuty)
- JIRA
- Custom (Web front end analytics w.Hadoop)
- Custom (Amazon cost analytics w.GoodData)
- Custom (Metrics Dashboard)
Many Tools Are Awful
YOU DECIDE
DEVOPS
DEVOPS

...what is it?
Traditional Dev and Ops

TIME TO DEPLOY MY APP TO PRODUCTION!

MY SERVERS ARE DEEEEEEEEEAD!!!
What is DevOps?

“A cultural and professional movement”

Adam Jacobs
What is DevOps?

“System administrators participating in an agile development process alongside developers and using many of the same agile techniques for their systems work.”

The Agile Admin Blog

@ernestmueller  #Agile2014  @iteration1
Pillars of DevOps

CULTURE

AUTOMATION

MEASUREMENT

SHARING

By:

John Willis

Damon Edwards
Scrumming away....

Agile Team
Ready to deploy...

SYSADMIN CAT

DENIES YOU ACCESS
Metrics Promote DevOps

• How do you get the cat inside the circle? Herding cats is hard. Some people aren’t cat people.
• Metrics can be used to promote culture, understanding, and collaboration
• Metrics help keep those different disciplines in sync by providing tangible collaboration points
• MTTD, MTTR, performance metrics, events
• Bringing all the discipline’s metrics together cover your whole value chain “code to cash”
Operations – Code = IT
IT + DevOps = ?

• Many IT teams implement Agile today
• They can implement DevOps too
• But to do either, they have to change how they interact with others
• Focus on customer’s needs not own needs; cloud/SaaS providing “competitive pressure”
• Practice Theory of Constraints – embed when possible, even if you need to add some
• Add devs and automate
EPIC RAP BATTLES OF HISTORY
Round Two

Small VS Large
SMALL

ORG
Typically 10’s of people

Everyone fits in the picture!
Fewer, less conflicting goals
One person: many hats!
Implement "Classical Devops"
No “scaling issues” yet!
So, what’s hard?
Do we need metrics?
Speed leads to confusion...
Build a culture of communication
Metrics 101: Culture of communication

• Talk in terms of metrics
  – Builds common ground between different roles.
  – Understand different perspectives.
  – Find the best way to get everyone talking in 1 place.
Metrics 201

• Push your metrics into your conversation tool
• Use tools that everyone likes:
  – IRC v/s Hipchat/slack
• Integrate your metrics into a channel
  – “Deployment channel” in your chat
Culture of communication

• Find a way to get people talking.
• Find face to face time with stakeholders.
• Metrics are that specific item to have a conversation around.
• Engineering teams love IRC, but business and PM’s might not as much.
• Transitioned to Slack/Hipchat (integrations and message history)
• Leads to visibility and builds trust
End Result

- Metrics drive conversations between everyone.
- Enhances productivity.
- Helped us streamline our process.
LARGE
ORG
Large Mature Org

- Hundreds of developers
- Many teams (many goals, processes)
- Distributed teams
- International teams
- Outsourcers
- Various Weird Partner Relationships
Large Org Problems

• Silos Galore
• Communication Problems
• Annoying Compliance Requirements
• Profitability Actually Important
• Less pure greenfield work – also responsibility for many existing mature systems
Story Time

- Story Time: SaaS product, 40 Engineers, 2/3 outsourced, mostly maintenance but extreme scale (1/3 of staff were Ops)
- Lots of support initiated urgent customer requests
- Dev still required for features, integration/transition with newer services, bug fix, scaling
- Team morale issues
Metrics 101

- First, add Agile. (Previously the ‘stew method’)
- Basic Metrics – number of tickets (100+ in queue at any time), size of backlog (500 or so bugs and stories), rate of new inflow and completion.
- Used to fix misunderstanding from upper management and correct resourcing.
- Next step on metrics – how to balance the support work and new work?
Metrics 201

PRR SLA Dashboard

93%

of tickets are in SLA

161
128
8
24
1
3
16%
92%
54

SLA tickets
closed in SLA
closed out of SLA
open in SLA
open out of SLA
3 day warning
% increase (vs Q1FY13)
target SLA
days left in quarter

Support SLA
Metrics 201

- Balancing these two metrics was the key to satisfying customers short and long term.
- But it’s not an either-or – by seeing the effects of people, process, and technology changes on those metrics we drove SLA from <50% to 100% and kept velocity growing (20.. 50... 200...)
- Having the metrics to focus on gave shared purpose and eased communication with the large distributed team
- Experiment, see the impact, pivot.
Metrics 301

- Monthly “Operational Excellence (Metrics) Meeting”
- Teams presented their metrics portfolio – with some variation as appropriate
- Drawn from system info, app metrics, db reports, Salesforce, surveys, etc.
- Keep it lean!!!

- Revenue and Cost
- Product Usage
- Performance
- Availability
- Client Satisfaction
- Employee Satisfaction
- Quality
- Security
Metrics 401 – A/B Testing

• All features had usage measured
• Feature flags would turn features on for customer subsets to measure usage, effect on conversion, etc. before committing
• Sometimes you had to kill it despite work spent
• Retooling could save a high profile failure
• “Yes, product guy, you have to.”
• Look for things metrics say you can kill – it’s the only way to stay lean long term
YOU
DECIDE
KANBAN
SCRAM
IT'S A TRAP
Here’s why...

- How many meetings?
- “Short planning meeting”?
- How often do these go long?
- Wait how long before prioritizing a feature/bug?
- Role of a dedicated scrum master is a luxury.
- Derailed sprints because of changing business priorities...

@ernestmueller  #Agile2014  @iteration1
Why Kanban?

- Limited number of WIP tasks in play.
- Easier to prioritize. There is only 1 list!
- Standups are simpler.
- Task estimates in days versus hours (1/2 day → 7 day).
- Research tasks to figure out how long something may take.
Kanban benefits

• Kanban + CI == Solved our issue of “when to release”. Didn’t have to wait for release windows like in scrum.

• Less stressful == Only x number of tasks going on at once. Easier to measure.

• Velocity is awesome!
Kanban Metrics

• Things we track:
  – Visualized board (JIRA Greenhopper)
  – Cycle Time (How fast something gets done)
  – WIP (Work/Tasks in progress)
  – Flow diagram
  – % of bugs
SCRUM
Scrum Rules

• Have used Scrum for both pure Ops and Dev+Ops teams
Kanban Drools

- Deadlines help maintain tempo – we had multiple releases a sprint, don’t need to tie them together
- You can reliably commit to a near term ETA with Scrum instead of just “when it’s done”
- Scrum has a better backlog (esp. in JIRA!)
- Many people “doing Kanban” are really “doing nothing”, like some doing “Agile” are really doing “cowboy coding.” Kanban takes more discipline and training than Scrum.
Scrum and Metrics

- Velocity is easier for people to understand than flow diagrams
But I Hear Kanban Is Better For Ops

• In a DevOps world, most Ops work SHOULD NOT be interrupt driven – it’s project work just like the devs are doing

• Dev and Ops expedite work approach each other in magnitude over time assuming appropriate investment in automation

• You may be thinking of “Level 1 Support” or “The Helpdesk” – that is NOT an Ops Engineer
Scrum for Ops?

- Devs have to be involved in major incidents too!
- Over the length of a sprint, the interrupt level evens out – my metrics show that velocity doesn’t vary more than with dev teams
- You manage WIP in your scrum too
Complications Scrum Helps

- Distributed teams need more communication ceremonies
- Foreign/contract workers need more communication ceremonies
- Same process across teams is better – in most cases other teams were using Scrum
- Simple common metrics → better collaboration
- When starting from zero, Scrum was the quickest path to team continuous improvement
YOU
DECIDE
Too Many Metrics

ONE HUNDRED BILLION METRICS
Cargo Cult Metrics
Demand Perfection
Weaponized Metrics
Recap

• Metrics are good – use them, be guided by them, communicate with them.
Recap

• Metrics can enhance your:
  – Culture
  – Productivity
  – Process
Recap

- Use them for good, not for evil.