DUBLIN
DRUPALCON

PEOPLE METRICS:

How to Use Team Data to Produce Positive Change

Amin Astaneh, Acquia Inc.

DevOps Track

Who am I?

Senior Manager, Infrastructure Services at Acquia

- Was in Operations Team from Dec 2010 - Nov 2015
- Formalized Incident Response and Ticketing Process
- Wrote automation tools to manage a rapidly-growing fleet (now ~15000)
- Implemented Kanban process in Apr 2015 to manage Ops work-in-progress
- Currently tech lead for Ops Tools Team, people manager for Tier2 Operations (soon-to-be SRE)

Acquia
THINK AHEAD.

So.. METRICS.

TouchWeb Live
Top page views / min (stacked)
Traffic & Response time
New visits (sessions) / min
Errors (500) / Min
HTTP 200
API msecs (articles)
API msecs (subscriptions)
API msecs (folders)
Memory used
Memory free
Memory %
RSS feeds queued
New RSS feeds
New articles
RSS feed errors
Network recv (bytes)
Network recv (errors)
Network recv (dropped)
RSS feeds retrieved
Total RSS feeds
Total articles
RSS feeds skipped
Network sent (bytes)
Network sent (errors)
Network sent (dropped)
IO read (bytes)
IO read (time)
IO write (bytes)
IO write (time)
Disk space %
CPU %
Load avg
Dashboards
My GroundWork
Event Console
Status
Views
Reports
Auto Discovery
Configuration
Administration
Advanced
Resources
Events : Entire Network
Host Status: Entire Network
Service Status: Entire Network
Monitoring Statistics : Hosts
Monitoring Statistics : Services
Troubled Hosts: Entire Network
Troubled Services : Entire Network

So.. METRICS.

What do we usually think about when it comes to ‘metrics’ for a product or service?

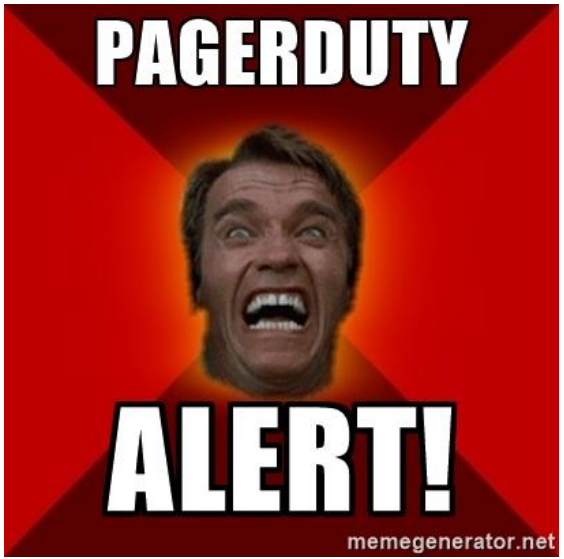
- Utilization
- Saturation
- Availability
- Latency
- Error Rates
- Throughput

- Hardware (CPU, Memory, Disk, Network)
- OS (Network Connections, Open Files)
- Services (Number of requests, cache miss/hit)
- App-Level (HTTP responses, clickthroughs, sales, etc)

So.. METRICS.

We use this information to drive decisions around:

- Should a person get paged?
- Do we need to scale our infrastructure?
- Do we need to revert that last deploy?
- Should we keep that feature?

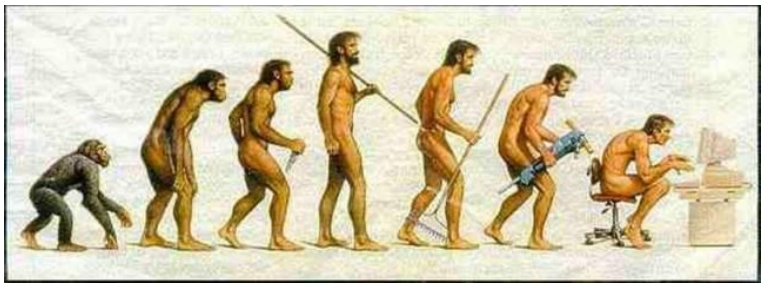


DUBLIN
DRUPALCON
It’s NOT the Whole Picture!

Humans build and operate software. They are an essential piece of the mechanism that keeps a service up and customers happy.

Therefore:

It stands to reason that we should be measuring them, too!



What Can 'People Metrics' Accomplish?

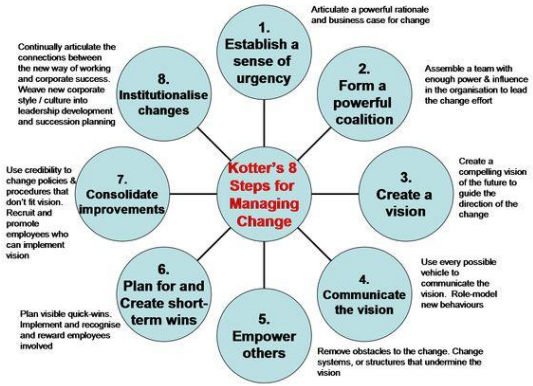
If you are a **manager** trying to keep your team engaged, happy, and retained, such metrics can enable you to:

- be **proactive** about quality-of-life issues (alerts fatigue, toil, etc)
- make team status **transparent** to the rest of the company
- make **justification** for additional funding for staffing/resources
- identify **opportunities** for process improvement

What Can ‘People Metrics’ Accomplish?

If you are trying to **raise urgency** around an opportunity/problem in your organization, ‘people metrics’ can:

- convert **anecdotal** experience into **empirical** data
- reveal the **operational cost** of current conditions to leadership
- identify **constraints** in key business functions
- **win members of leadership to your cause**



http://www.kotterinternational.com/the-8-step-process-for-leading-change/
https://en.wikipedia.org/wiki/Theory_of_constraints

What Can 'People Metrics' Accomplish?

If you are a **business leader**, people metrics can:

- **Quantify** the level of **efficiency** your teams have in **creating value**
- **Identify** where organizational **pain points** are
- Be equipped with the essential data necessary to make **tactical decisions**
- **Ensure Customer Success**

Simply complaining about a problem isn't going to work.

DUBLIN
DRUPALCON
“The goal of an organization is to increase **throughput** while reducing both the **inventory** and **operating expense**.”
– Eli Goldratt, “The Goal”

DUBLIN
DRUPALCON
YOU HAVE TO COMMUNICATE WITH LEADERSHIP IN THOSE TERMS!
(throughput, inventory, operating expense)

What can ‘People Metrics’ accomplish?

What will influence decision makers more effectively?

“Working on Team X stinks. We are always firefighting and doing tickets.”

“40% of Team X’s time is spent on incident response, and 30% is spent on manual tasks that the business needs. That is 70% of their time not spent on making improvements to the product or streamlining current processes.”

Metric: Time/Effort Spent in “4 Types of Work”

The Phoenix Project posits that there are four types of work in IT Operations. I argue the same is true for development teams too!

- **Business Projects**: new features
- **Internal Projects**: cleaning up tech debt, investment in CI/CD
- **Operational Change**: releasing, provisioning, configuring
- **Unplanned Work**: outages, firefighting, etc.

If we measure the quantity and percentage of each type of work over time, the business can know where their money is being spent and ensure maximum return-on-investment.

Metric: Time/Effort Spent in “4 Types of Work”

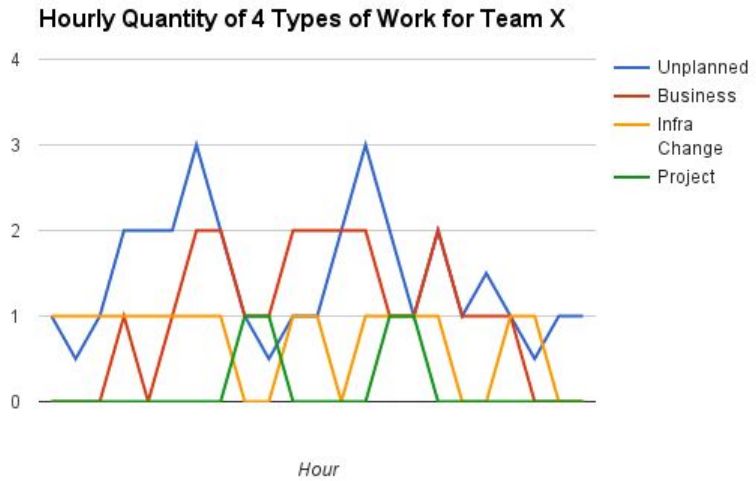
What can one do with such data?

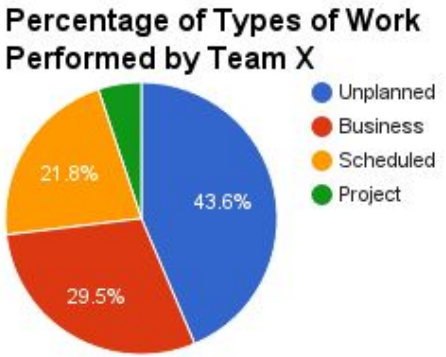
- You can make decisions to keep unplanned work to a minimum.
- For a development team, you can target for maximum time spent on business projects (new features)

  (business > internal > ops change > unplanned)
- For an operations team, you can target for maximum time spent on internal projects (make the service reliable and automated, streamline manual tasks)

  (internal > business > ops change > unplanned)

Metric: Time/Effort Spent in “4 Types of Work”

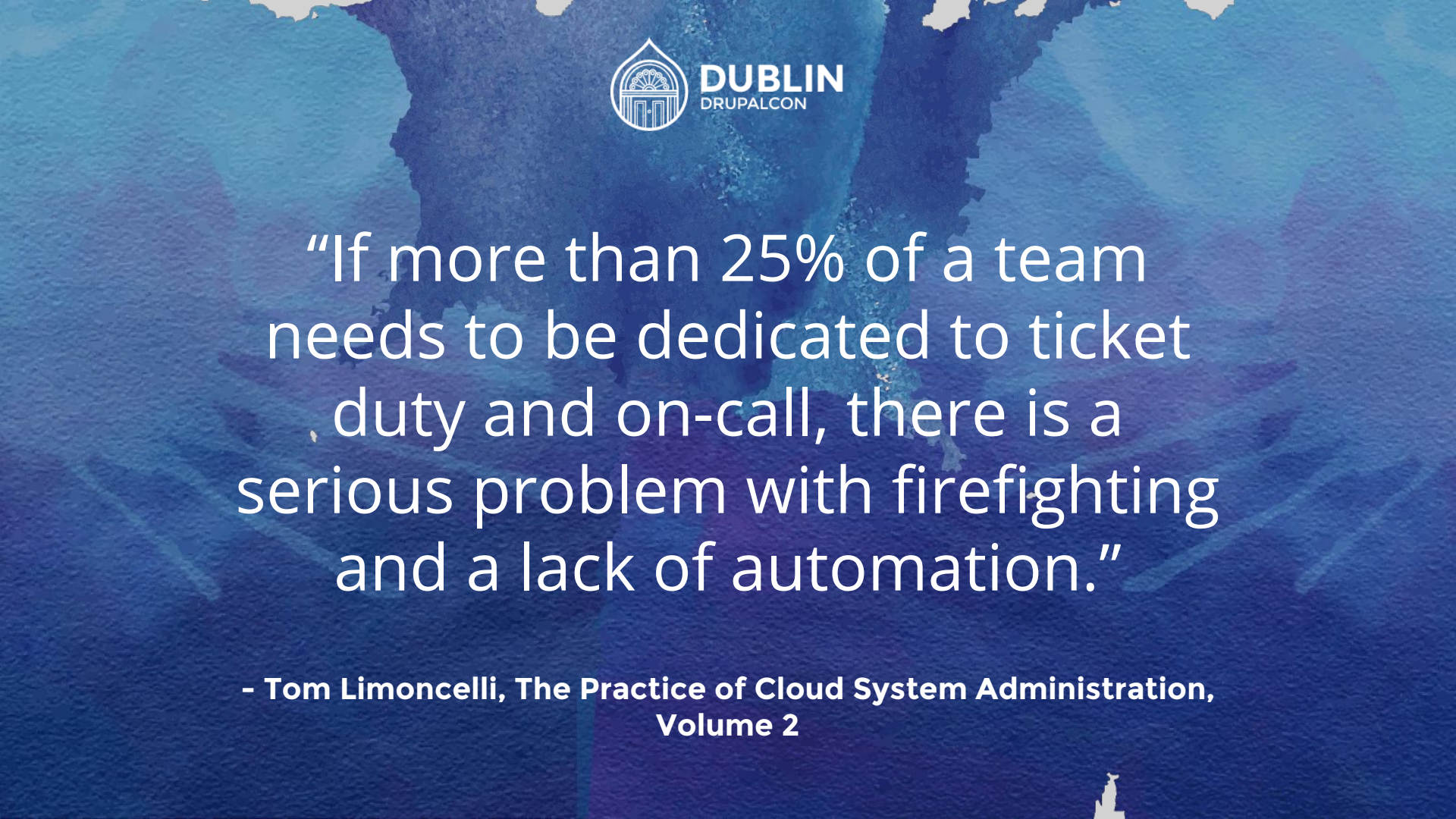




How healthy was this team today?

DUBLIN
DRUPALCON

UNPLANNED WORK IS WASTE


DUBLIN
DRUPALCON
“If more than 25% of a team needs to be dedicated to ticket duty and on-call, there is a serious problem with firefighting and a lack of automation.”
- Tom Limoncelli, The Practice of Cloud System Administration, Volume 2

A Simpler Metric: Operational Load

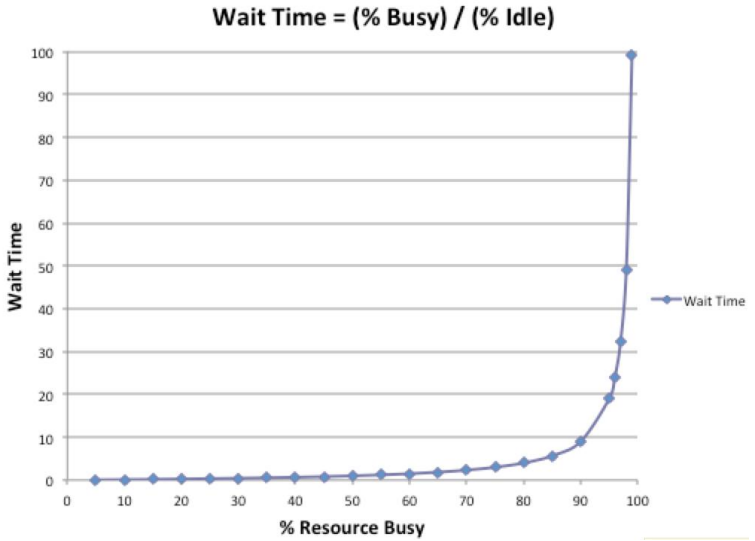
Operational Load is the percentage of time spent towards the upkeep of your service. **It's time not writing code or making improvements.**

Google caps this time at 50% for their Site Reliability Engineers. When exceeded, the ops work overflows to the software engineering team.

A Simpler Metric: Operational Load

Why 50%? Remember the wait time graph from The Phoenix Project?

- Once you exceed 50%, customers will start to wait longer for work to get done.
- As you approach 80% and beyond, it really gets out of hand.



DUBLIN
DRUPALCON
SLACK IS YOUR FRIEND
(no, not the chat service)

Slack Is Your Friend

Slack is simply a term for ‘idle time’.

Having slack means your team can be responsive to bursts of unplanned work without a business impact.

Slack means opportunities to improve skillsets and morale.

The 20’th Century management style of keeping slack lean/nonexistent **doesn’t work** (that creates constraints!). Flow of work can be inconsistent. Be Prepared!

Metric: Happiness

Every $INTERVAL, ask your team these questions:

From a scale of 1-5:

- How happy are you doing your job?
- How happy are you working at your company?

Also:

- What makes you the most happy?
- What makes you the least happy?
- What single thing, if changed, would most greatly increase your happiness?

Metric: Happiness

What can this metric do for you?

- Quantify common morale of the team (effects of toil, crisis, etc)
- Identify common improvement opportunities
- Prevent burnout, employee turnover, etc.
- Allows for a safe place for people to sound off on team issues (especially if you allow anonymous submissions)

EMPLOYEE TURNOVER IS EXPENSIVE!

Other Metrics

- Cycle Time: how long will a customer wait on a request?
- Throughput: requests performed per day/week/month
- Frequency By Request Type: what should be automated first?
- Frequency By Root Cause: what is causing the most pain?
- Reopened Issues/Bugs: how often are defects going downstream?
- Time Spent Per Customer: is a particular customer profitable to keep?

How Do I Get Started?

1: Track Your Work in a Ticketing System

- It's Question 1 of the Ops Report Card for a Reason! http://www.opsreportcard.com/section/1
- In order to get accurate metrics, all work for your team needs to be tracked there.

2: Log Time Spent for Every Issue

- Ops/SREs should track ALL of their time
- Developers should track time spent performing non-coding tasks

But Tracking Time Sucks!

But Tracking Time Sucks!

- **Yes, it does.** It still needs to happen.
- SaaS tools like Toggl make it easier
- Writing tools that integrate with your ticketing system make it easier
- Emphasize over and over why time tracking is important
- Provide incentives to accurately track time
- **NEVER** use time tracking data as a weapon

3: Track Non-Issue Data Using Custom Tools

- Time-Series Databases like StatsD/Graphite, InfluxDB are VERY useful
- Worst case scenario: Use Google Forms!
- More on this later

4: Make Dashboards and Make Them Visible

- Grafana is *very* useful for this
- If using Jira, widgets can be used to make a dashboard
- Display them in a prominent space in your office
- Document what the data means!

The goal is to generate EMPATHY for your team's current state.

5: Interpret and Communicate the Data

- Review them daily/weekly as part of your standups
- Ask questions and dig into the ticket system to find root causes for the team's current state
- Be able to articulate information in the form of a story, eg: "The recent code push caused X hours of unplanned work this week, which resulted in a reduced ticket throughput by Y%."
- Share with management

How Do I Share this Data with Management?

Again, it's all about operational cost, inventory, and throughput, so speak in terms of **TIME** and **MONEY**.

- **$5000** of Team X's time was spent rebooting servers due to Bug Y.
- Customers are waiting up to **2 weeks** for Team X to fulfill requests.
- It takes **one hour on average** to perform Task X.
- We need **double our usual EC2 costs** while Bug X is unresolved.

6: Define a ‘Target Condition’ and Set Goals to Achieve It

Create Target Conditions to improve an aspect of your team’s performance that can be expressed by a metric, a specific value, and a duedate. Then simply use the scientific method until the goal is achieved.

- Reduce operational load to < 50% in 6 months
- Reduce 90th percentile cycle time on tickets to 1 week in 3 months

(https://www.amazon.com/Toyota-Kata-Managing-Improvement-Adaptiveness/dp/0071635238?ie=UTF8&*Version*=1&*entries*=0)

No, Seriously: Show Me How to Create Metrics!!

Quick and Dirty Happiness Metrics

This example uses StatsD gauges:

```
#!/bin/bash
read HAPPINESS
echo "team.$(whoami).happiness:$HAPPINESS|g" \
| nc -w 1 -u statsd.server.tld 8125
```

Quick and Dirty Interruptions Tracking

This example uses StatsD counters:

```
#!/bin/bash
echo "team.$(whoami).interruptions:1|c" \
| nc -w 1 -u statsd.server.tld 8125
```

Demo Grafana Personal Dashboard

Let’s run these tools and see what happens!

But I Can't Code!

No problem!

- Jira has many reporting capabilities (built-in and with plugins)
- Business Intelligence Tools (Domo, Amazon Quicksight)
- Google Forms! (no, seriously..)

Google Forms Demo

You can create a decent happiness metric form in a matter of minutes using Google Forms. Don’t believe me? **Let’s do it right now.**

About Jira..

The JIRA API doesn't have everything you need. Here's some tips if you want to mine for goodies in the database:

If you assume that a set of comments in a ticket for a given day should have time tracked, you can then audit for missing time log entries. The tables you should care about are **worklog**(time tracking) and **jiraaction**(comments).

About Jira..

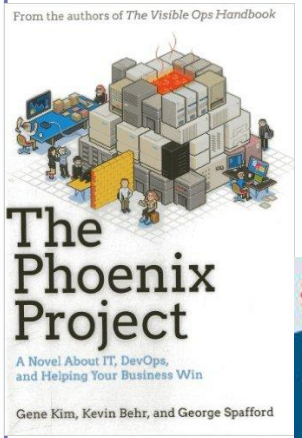
Jira doesn't really have good functionality for creating time-tracking reports. But...

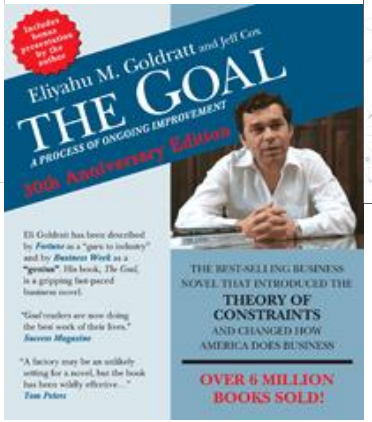
- Create a custom field called "Work Type" with values 'Business', 'Internal', 'Ops Change', and 'Unplanned'.
- In the Jira database, join worklog.issueid against customfield.ISSUE and look for specific customfieldvalue.CUSTOMFIELD values
- Using the SUM operator will allow you to aggregate time spent over desired timeframes for specific types of work
- Push the data to a time-series database

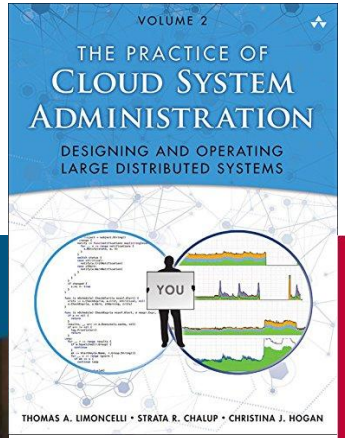
(https://developer.atlassian.com/jiradev/jira-platform/jira-architecture/database-schema/database-custom-fields)

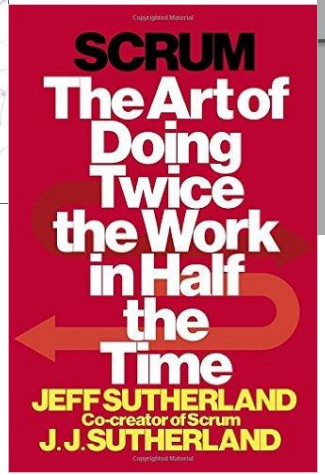
DUBLIN
DRUPALCON

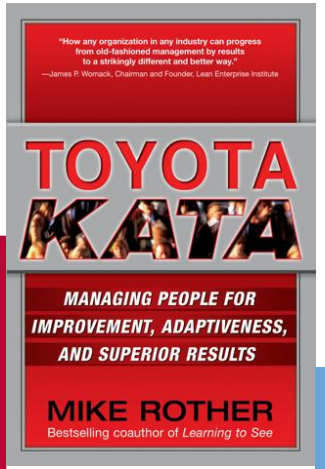
References

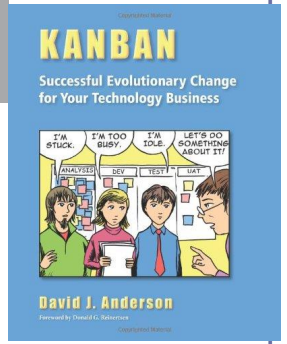












References

- The Phoenix Project
- The Goal
- The Practice of Cloud System Administration, Volume 2
- Scrum: The Art of Doing Twice the Work in Half the Time
- Toyota Kata: Managing People for Improvement, Adaptiveness, and Superior Results
- Kanban: Successful Evolutionary Change for Your Technology Business
- https://github.com/kamon-io/docker-grafana-graphite

DUBLIN
DRUPALCON

Thank You!

Amin Astaneh
Twitter: @aastaneh
Freenode: amin

DUBLIN
DRUPALCON
WHAT DID YOU THINK?
Evaluate This Session
events.drupal.org/dublin2016/schedule
THANK YOU!