Jeff Einhorn
@jeffeinhorn

innovation in the **LARGE** enterprise

hello

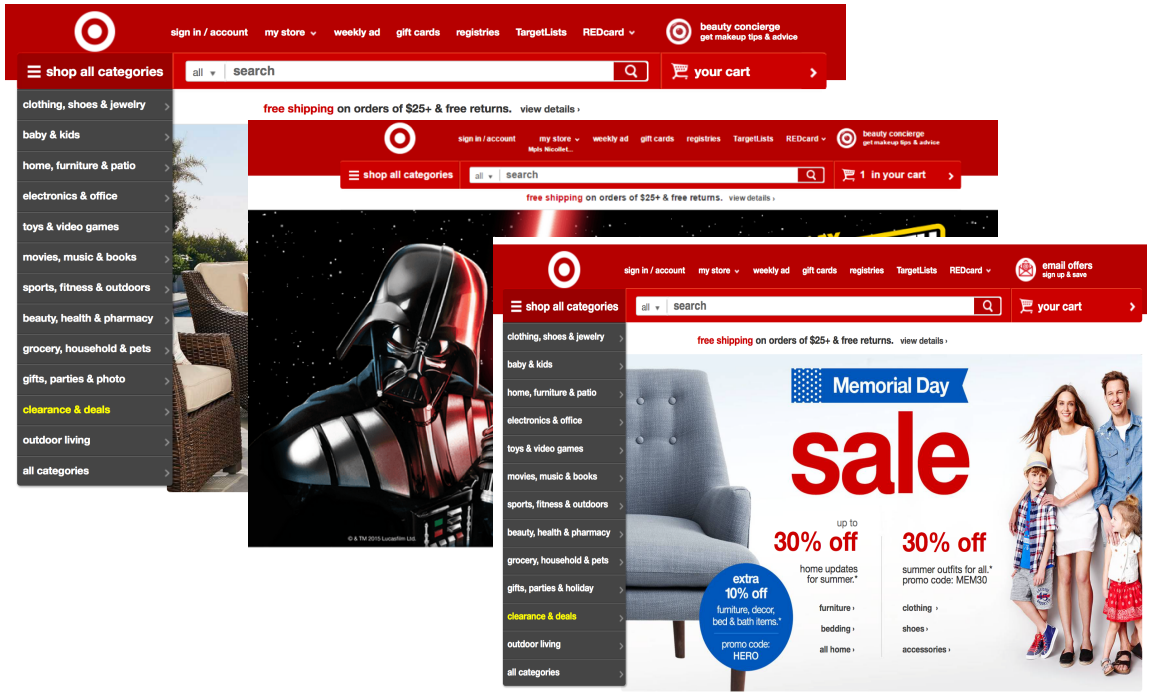
+347,000 team members

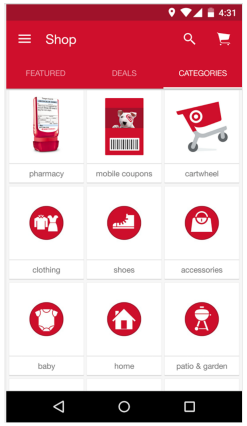
1,795 stores in 49 states

41 distribution centers

7 HQ locations

2 data centers





ideas can sprout from a number of different sources including things you've

- seen
- heard
- read
- learned
- experienced

inspiration

inspiration

get, stay connected with the broader tech community

inspiration

CHEFCONF2015
GAME ON
CHEF

RED HAT SUMMIT
redhat

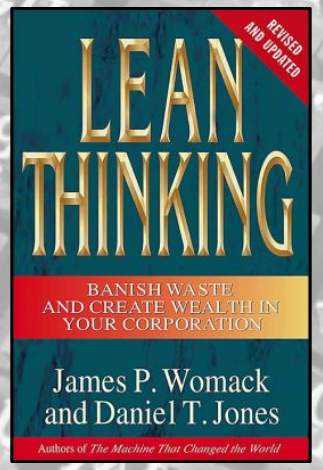
O'REILLY
Velocity

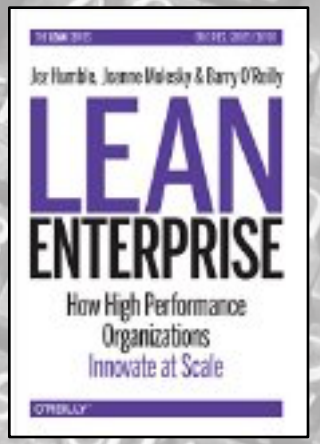
DEVOPS DAYS

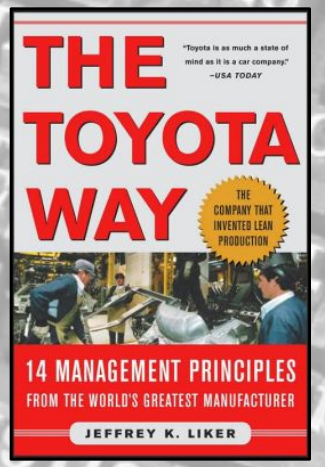
Monitorama
An Open Source Monitoring Conference & Hackathon

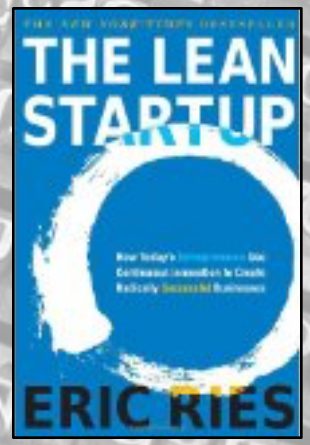
inspiration

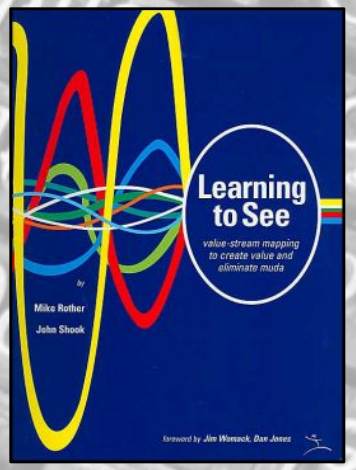
continuous learning is critical to being able to generate new ideas, refine existing work, or kill an afternoon

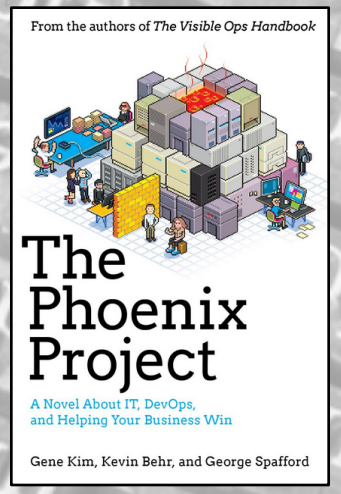


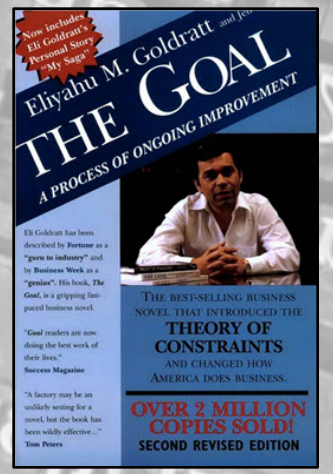




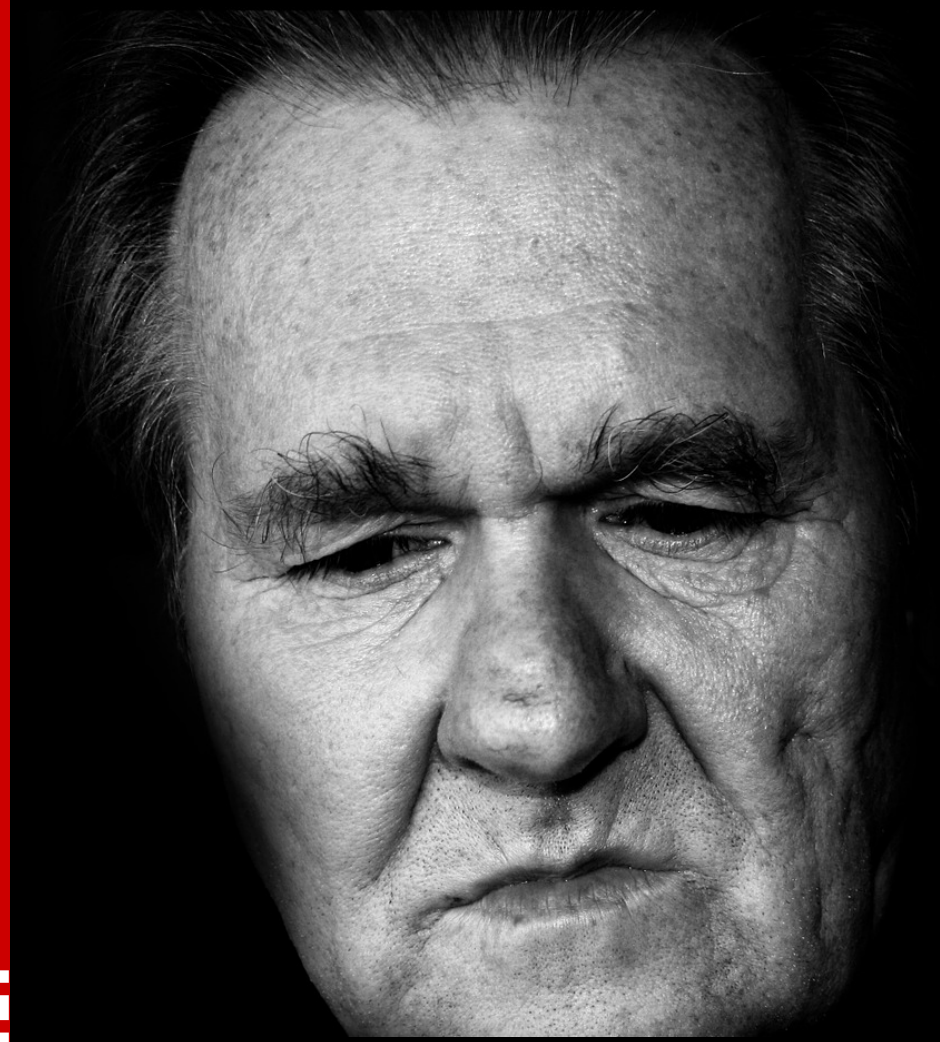




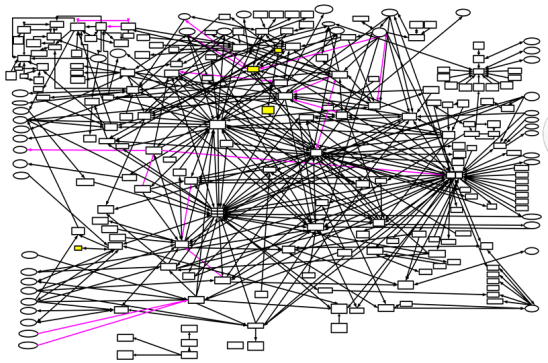


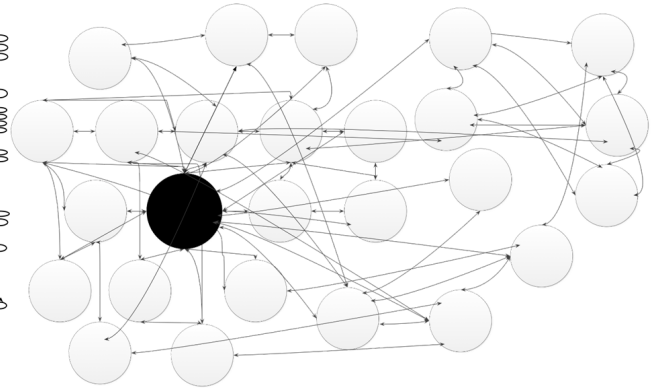


inspiration

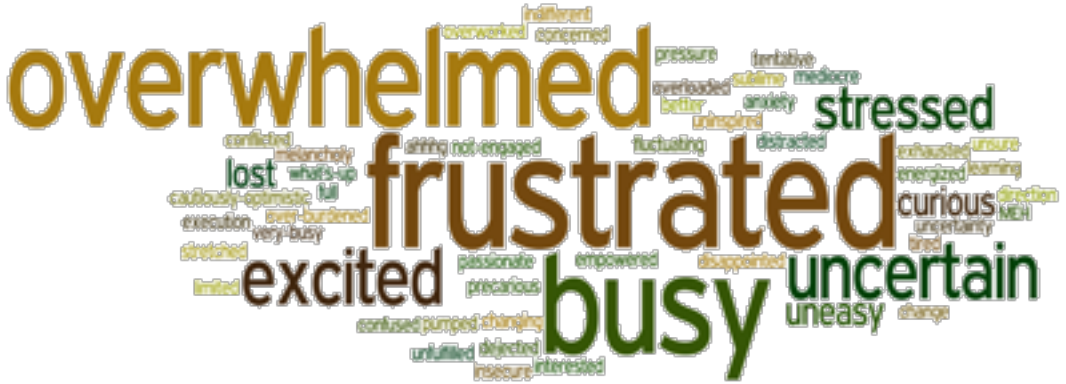


what's driving you, your team, your clients, or the business crazy?





optimization **HAIRBALLS**



getting focused

spending too much time putting out fires and only getting more and more behind

	urgent	not urgent
important	**QUADRANT 1** ▪ Crises ▪ Deadline driven projects ▪ Fire-fighting	**QUADRANT 2** ▪ Building capabilities ▪ Maximizing opportunities ▪ Risk management
not important	**QUADRANT 3** ▪ Interruptions ▪ Most meetings and e-mail	**QUADRANT 4** ▪ Trivia ▪ Busy work ▪ Time wasters

Stephen Covey's approach to time management is to create time to focus on important things before they become urgent. Sometimes this just means doing things earlier. The real skill is to commit time to processes that enable you to do things more quickly or more easily, or ensure that they get done automatically.


the solution to everything
that's wrong in your life
DevOps
help!

but .....

DevOps seems to describe a lot of things

- culture
- tools
- technology

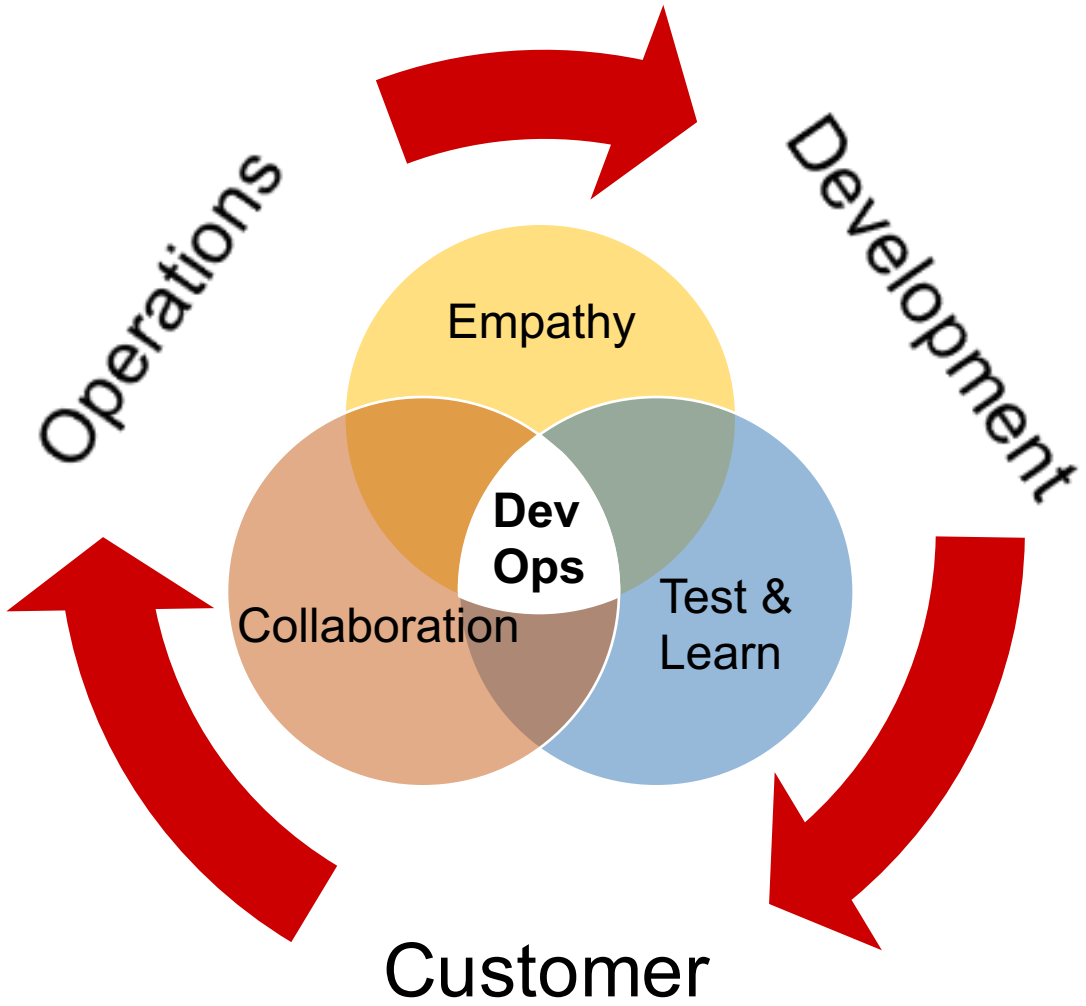
what *is* it?

DevOps@TGT

Culture

Ownership

Empowerment



Feedback

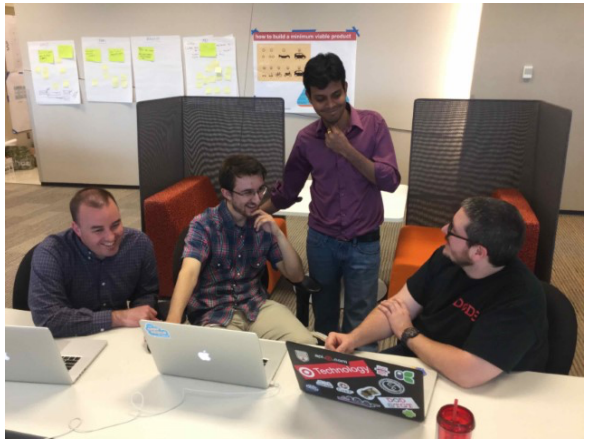
Tech Practices

Measurement

culture

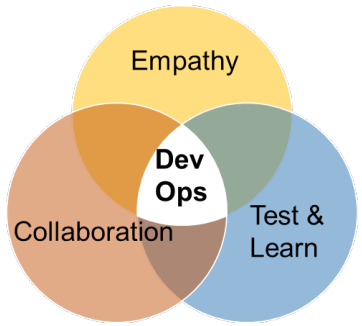
Shared Goals versus Operations = Stability & Development = Features

Operations & Development engineers participating together in the entire service lifecycle, from design through the development process to production support



what it looks like @ Target

- Open Labs
- Challenges
- FlashBuilds
- Coaching
- #ChatOps
- #HugOps
- DevOpsDays
- Leader Summits



community

products & services

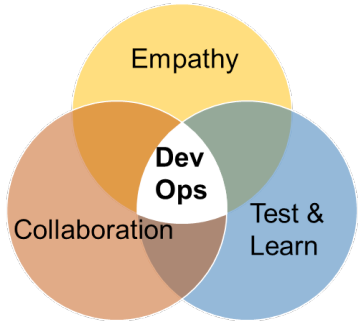
ownership

Product/Service Ownership versus Project

Products & Services are delivered, continuously improved and managed for the long term by end-to-end by full stack teams focused on meeting customer needs

what it looks like @ Target

- Project Argus
- *real* Product ownership

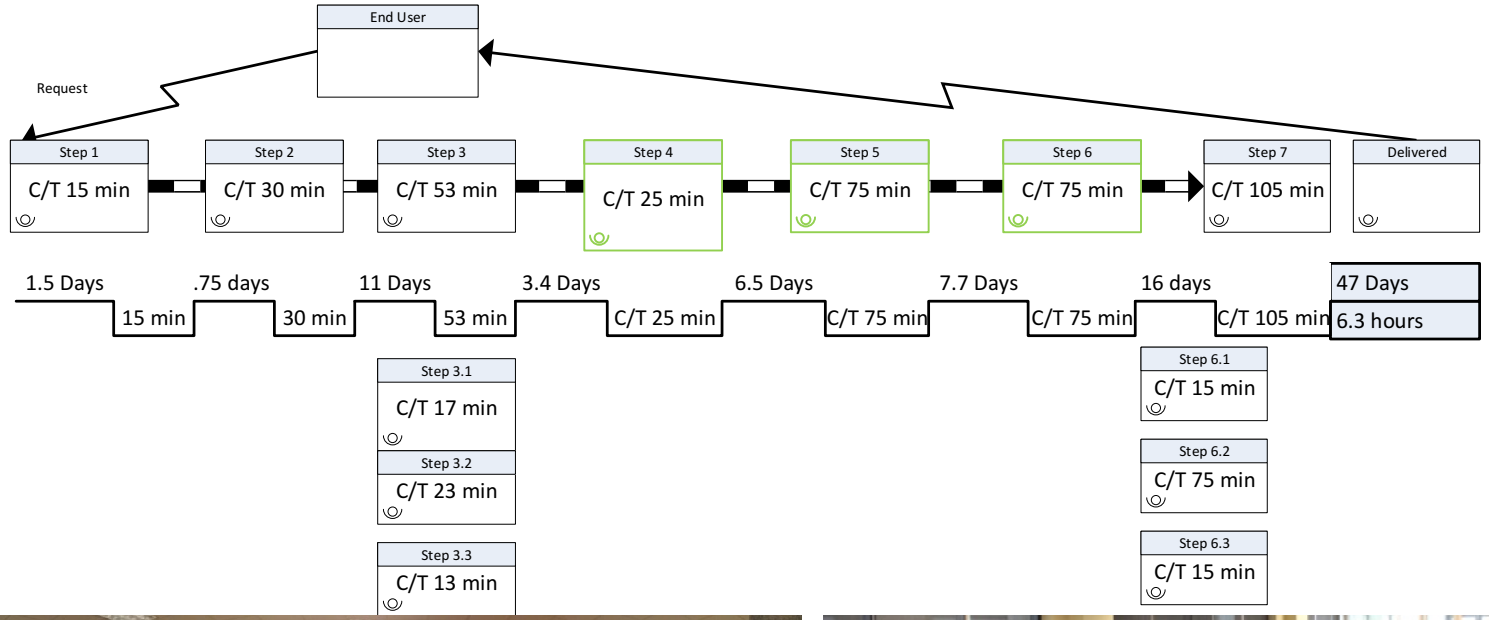


make work visible

measurement

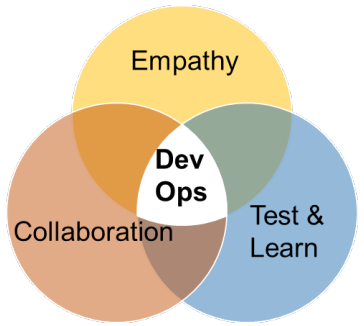
Sub-optimization versus Identifying Bottlenecks

Measure everything you want to improve. Value stream process before you start, measure deployments, time to deploy, incidents per deploy, blast radius/severity of incidents. Report on service adoption & competing service adoption to better understand customer needs and how to improve



what it looks like @ Target

- value stream mapping
- dashboards tied to business critical KPIs
- end-of-sprint demos

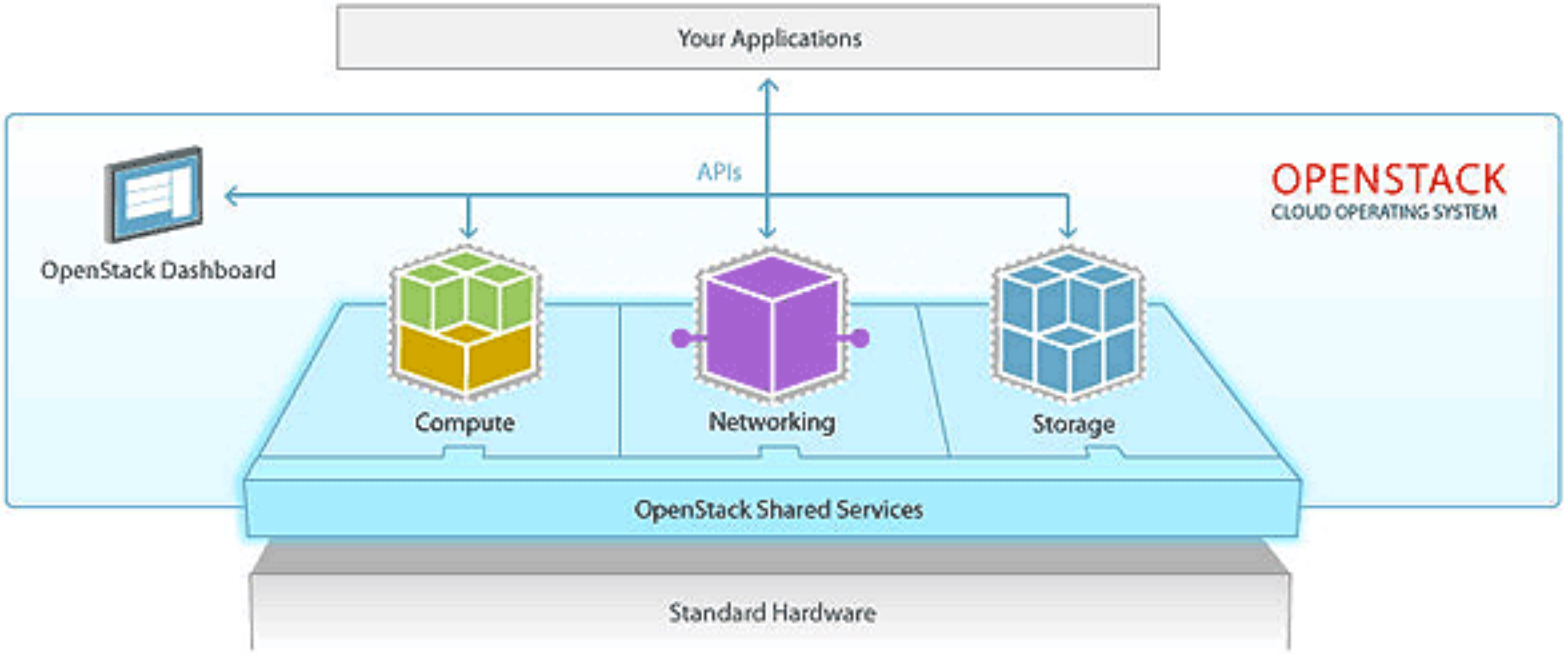


power to the people

empowerment

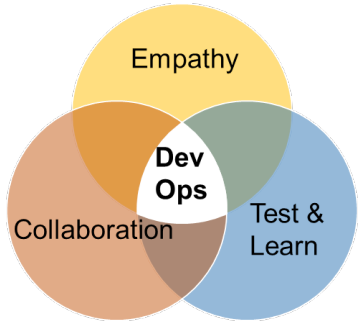
Automated Build & Deploy of App/Infrastructure versus separate engineers hand crafting pieces

Engineers have access to the tools they need and can improve existing processes to solve problems and improve delivery



what it looks like @ Target

- self-service api access to create
  - networks
  - servers
  - compute



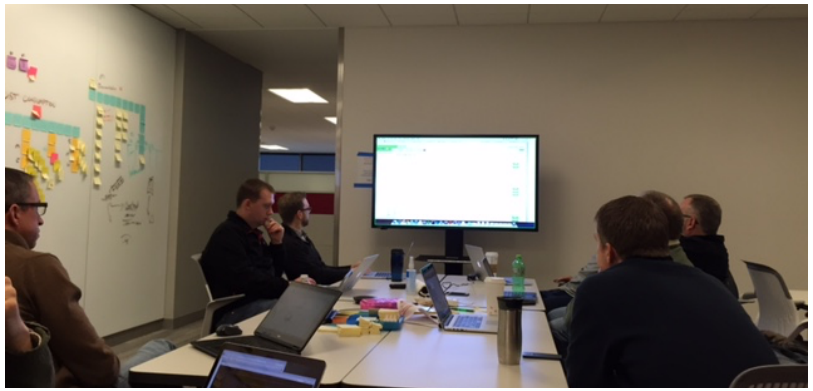
keeping in touch

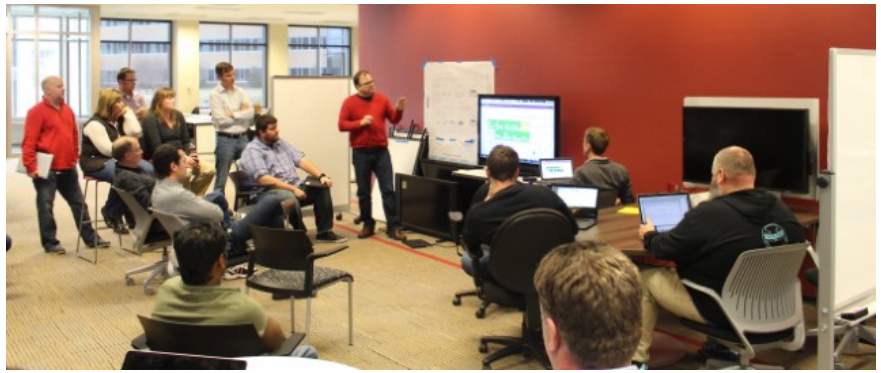
feedback

Frequent Small Changes versus Large Changes

Agile Sprints versus Annual Planning

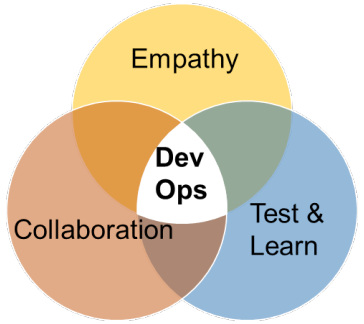
Operations & Development makes small changes frequently to learn quickly from failures & customer feedback then adjusts future works





what it looks like @ Target

- open demos (i.e., everyone is invited)
- consistent communication cadence
- #ChatOps
- strategic collocation

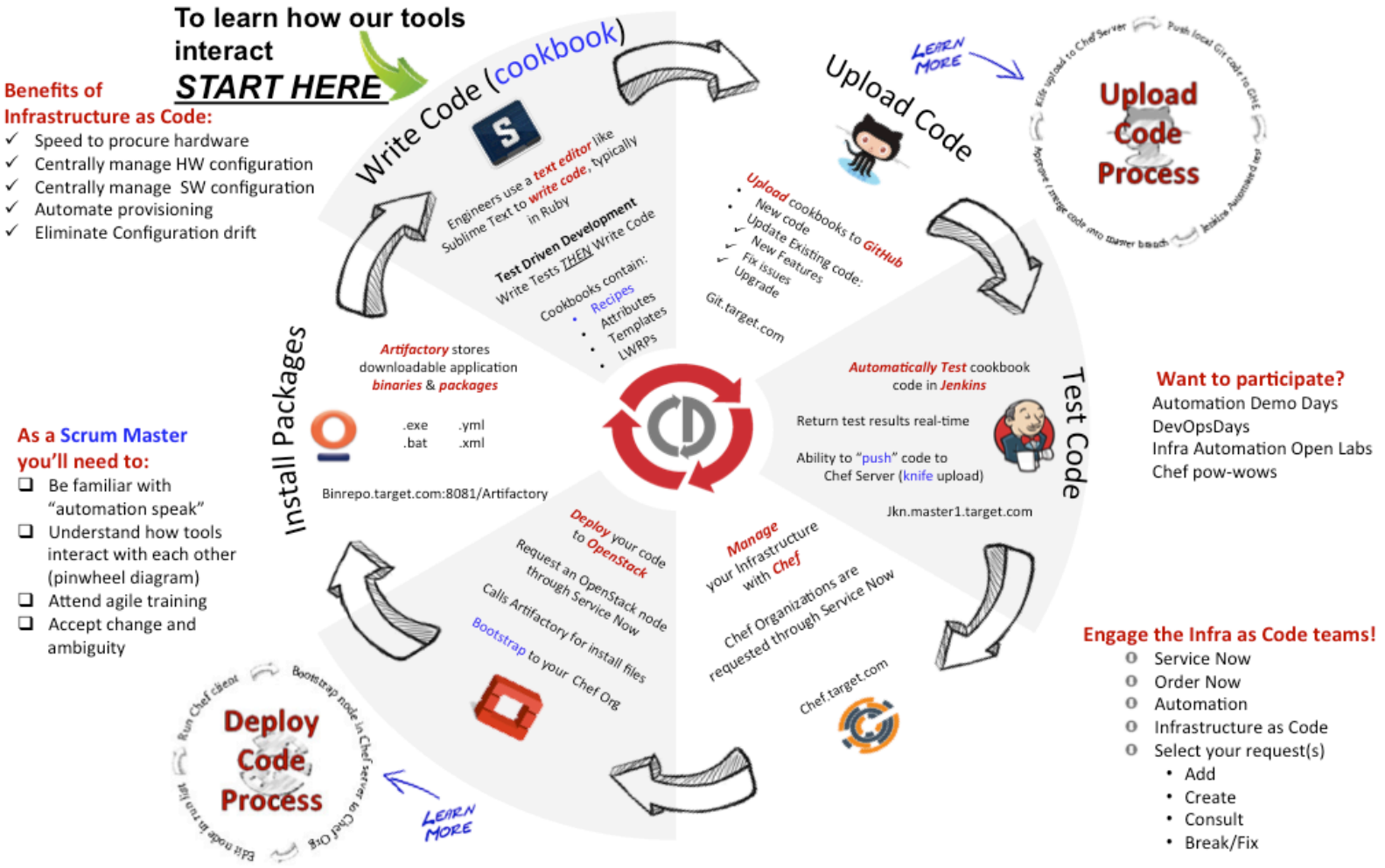


tech foundation

technology practices

Reusable Automation & Services versus Scripting & Monolithic applications

Infrastructure as Code, Shared Source Code, Social Coding, Agile Practices, Continuous Integration/Delivery, Scrum, Agile



what it looks like @ Target

- CI/CD tooling
- Scrum practices
  - sprints
  - release planning
  - standups
  - retrospectives
  - demos

DevOps@TGT

Culture

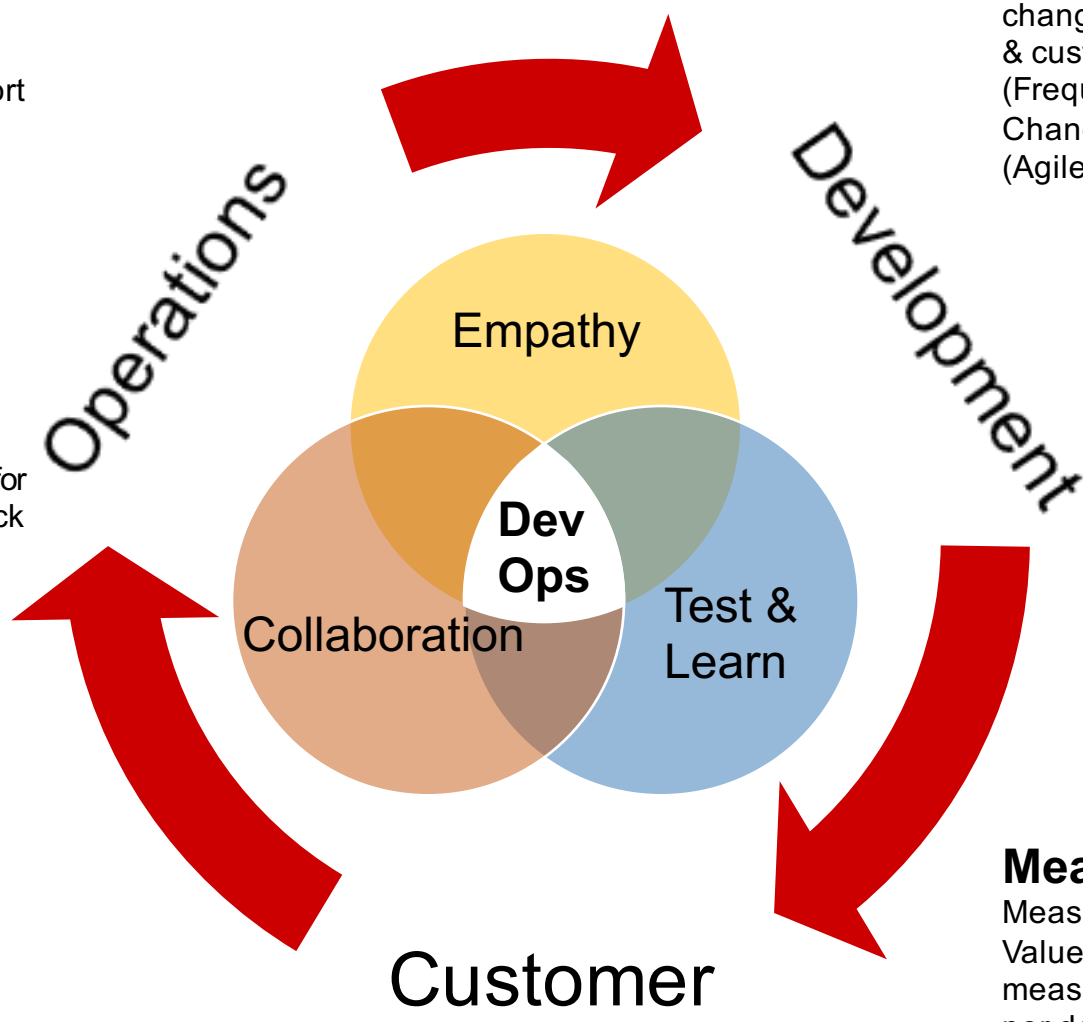
Operations & Development engineers participating together in the entire service lifecycle, from design through the development process to production support (*Shared Goals versus Operations = Stability & Development = Features)*

Feedback

Operations & Development makes small changes frequently to learn quickly from failures & customer feedback then adjusts future works (Frequent Small Changes versus Large Changes)
(Agile Sprints versus Annual Planning)

Ownership

Products & Services are delivered, continuously improved and managed for the long term by end-to-end by full stack teams focused on meeting customer needs
(Product/Service Ownership versus Project)



Tech Practices

Infrastructure as Code, Shared Source Code, Social Coding, Agile Practices, Continuous Integration/Delivery, Scrum
(Reusable Automation & Services versus Scripting & Monolithic applications)

Empowerment

Engineers have access to the tools they need and can improve existing processes to solve problems and improve delivery
(Automated Build & Deploy of App/Infrastructure versus separate engineers hand crafting pieces)

Measurement

Measure everything you want to improve. Value stream process before you start, measure deployments, time to deploy, incidents per deploy, blast radius/severity of incidents. Report on service adoption & competing service adoption to better understand customer needs and how to improve
(Sub-optimization versus Identifying Bottlenecks)

did I mention …

culture is important

it eats strategy for breakfast, lunch, dinner and as a midnight snack

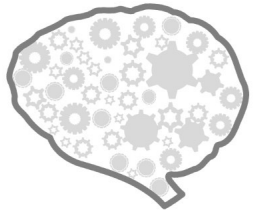
90%

estimate of organizations attempting to use DevOps without specifically addressing their cultural foundations will fail by 2018

http://www.gartner.com/newsroom/id/3032517

our growing methods

DevOps@TGT → internal social media site



→ internal DevOps mini-conference

automation make day → quarterly day long hackathon

automation demo spotlight → monthly session to share, get feedback, inspire & be inspired

case study

service management

hint: people look like this more often than not

why do this

empower customers to use service management for

IaaS

to get infrastructure on demand with all components, access needed to make things work

OpenStack

PaaS

to quickly prove out new ideas

OpenShift

Automation

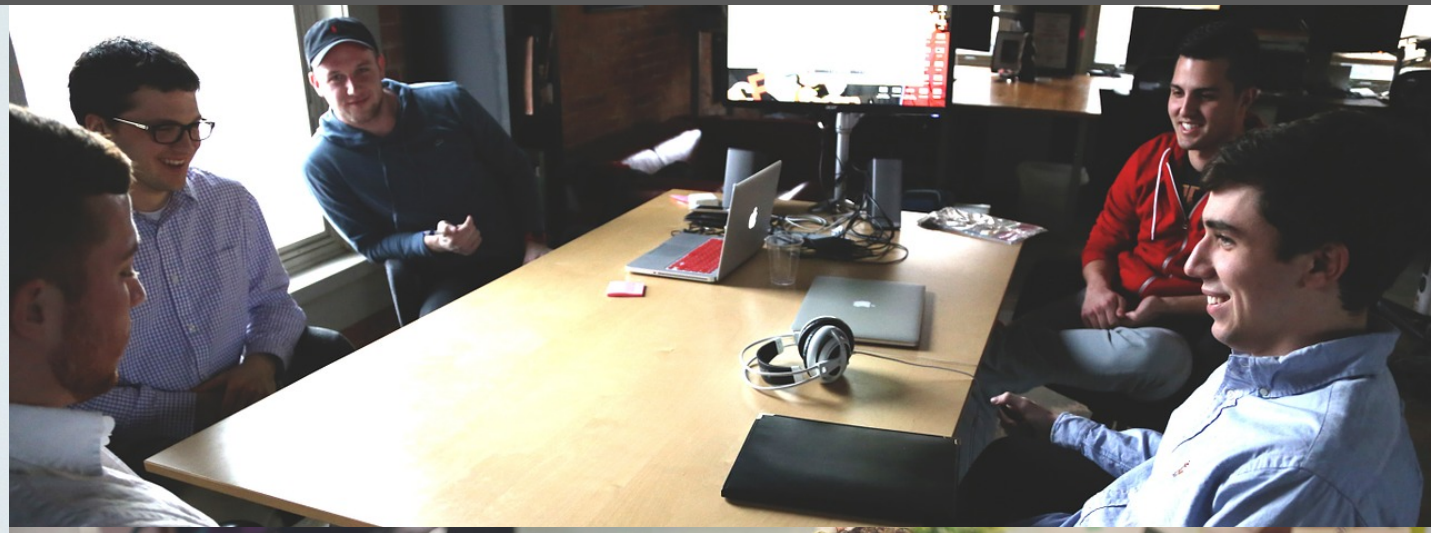
to create a foundation to build out services for everything in IT

Chef, Jenkins, etc.

step 2 | lay foundation

get what you need to succeed

hint: executive support to get the team, space and time needed to build out new capabilities



legacy approach

projects

long cycle times
siloes & hand-offs
fractured accountability
end state integration, testing
complex

modern approach

products/services

short cycle times
tight collaboration
end-to-end accountability
continuous integration, testing
simple

step 4 | make it real

IaaS pilot

legacy approach

3 months

modern approach

30 minutes

- identified software development team aligned to strategic priority
- provided team with
  - 1 embedded engineer
  - 1 part-time CI/CD coach
  - 1 part-time tool consultant
  - collocated work environment
  - IaaS + custom PaaS environment
  - CI/CD tooling environment
- team conducted 2-day sprints over 30 days with full Scrum ceremonies
- team worked directly with self-service infrastructure tools

PaaS pilot

legacy approach

8 weeks

modern approach

15 minutes

create development environment

- reduced number of steps (from 11 to 1)
- reduced number of requests (from 11 to 0)
- shift from Service Catalog to Self-Service Portal
- eliminate dependencies, pre-requisites, and hand-offs

Automation pilot

legacy approach

9 months

modern approach

10 minutes

create pipeline infrastructure

- reduced number of steps (from 13 to 2)
- reduced number of requests (from 13 to 1)
- shift from Service Catalog to Self-Service Portal
- eliminate dependencies, pre-requisites, and hand-offs
- use Chef to deploy Chef

signs of success

adoption is your measure of success

IaaS

1000+ VMs spinning up every day after 6 months

OpenStack

PaaS

Dozen's of developers via word of mouth and no support team

OpenShift

Automation

Thousands of Nodes & Cookbooks for everything. Checkout our community BigIP cookbook – 800k downloads

Chef, Jenkins, etc.

key learnings

be focused

be bold

pilot

iterate

ownership matters

#make_awesome_happen

you build it, you run it

Werner Vogels, Amazon CTO

get the right folks

set course

get leaders to buy-in


get stuff done

keep calm and scale on


make awesome happen

keep in touch

Jeff Einhorn

Twitter

@jeffeinhorn
#DOTGT

Target Tech Blog

http://target.github.io

The Goat Farm

http://goatcan.do