

What I Learned About Going Fast at eBay and Google

Randy Shoup

@randyshoup

[linkedin.com/in/randyshoup](https://www.linkedin.com/in/randyshoup)

GOTO Chicago, May 20 2014

Background

CTO at KIXEYE

- *Real-time strategy games for web and mobile*

Director of Engineering for Google App Engine

- *World's largest Platform-as-a-Service*

Chief Engineer at eBay

- *Multiple generations of eBay's real-time search infrastructure*

The Need For Velocity



Real-time strategy games are

- *Real-time*
- *Spiky*
- *Computationally-intensive*
- *Constantly evolving*
- *Constantly pushing boundaries*

Why Are Organizations Slow?

People

Organizational Culture

Process

Why Are Organizations Slow?

People

Organizational Culture

Process

People: Hire and Retain the Best

Hire 'A' Players

- *In creative disciplines, top performers are 10x more productive (!)*
- *Smaller, more productive teams*
- *Less management and coordination overhead*

Confidence

- *A players bring A players*
- *B players bring C players*

Google Hiring

Goal: Only hire top talent

- *False negatives are OK; false positives are not*

Hiring Process

- *Famously challenging interviews*
- *Very detailed interviewer feedback*
- *Hiring committee decides whether to hire*
- *Separately assign person to group*

→ Highly talented and engaged employees

People: Differences



Most valuable asset

- *Treat people with care and respect*
- *If the company values its people, people provide value to the company*

People are not interchangeable

- *Different skills, interests, capabilities*
- *We are not cogs, not fungible*

Create a *Symphony*, not a Factory

- *Beauty and richness comes from different instruments, playing together*
- *Compose teams to take advantage of differences*

eBay “Train Seats”

eBay’s development process (circa 2006)

- *Design and estimate project*
(“Train Seat” == 2 engineer-weeks)
- *Assign engineers from common pool to implement tasks*
- *Designer does not implement; implementers do not design*

→ Dysfunctional engineering culture

- *(-) Engineers treated as interchangeable “cogs”*
- *(-) No regard for skill, interest, experience*
- *(-) No pride of ownership in task implementation*
- *(-) No long-term ownership of codebase*

Virtuous Cycle of People



Why Are Organizations Slow?

People

Organizational Culture

Process

Organization: Quality over Quantity

Whole user / player experience

- *Think holistically about the full end-to-end experience of the user*
- *UX, functionality, performance, bugs, etc.*

Less is more

- *Solve 100% of one problem rather than 50% of two*
- *Users prefer one great feature instead of two partially-completed features*

Organization: Culture of Learning

Learn from mistakes and improve

- *What did you do -> What did you learn*
- *Take emotion and personalization out of it*

Encourage iteration and velocity

- *“Failure is not falling down but refusing to get back up” – Theodore Roosevelt*

Google Blame-Free Post-Mortems

Post-mortem After Every Incident

- *Document exactly what happened*
- *What went right*
- *What went wrong*

Open and Honest Discussion

- *What contributed to the incident?*
 - *What could we have done better?*
- *Engineers compete to take personal responsibility (!)*

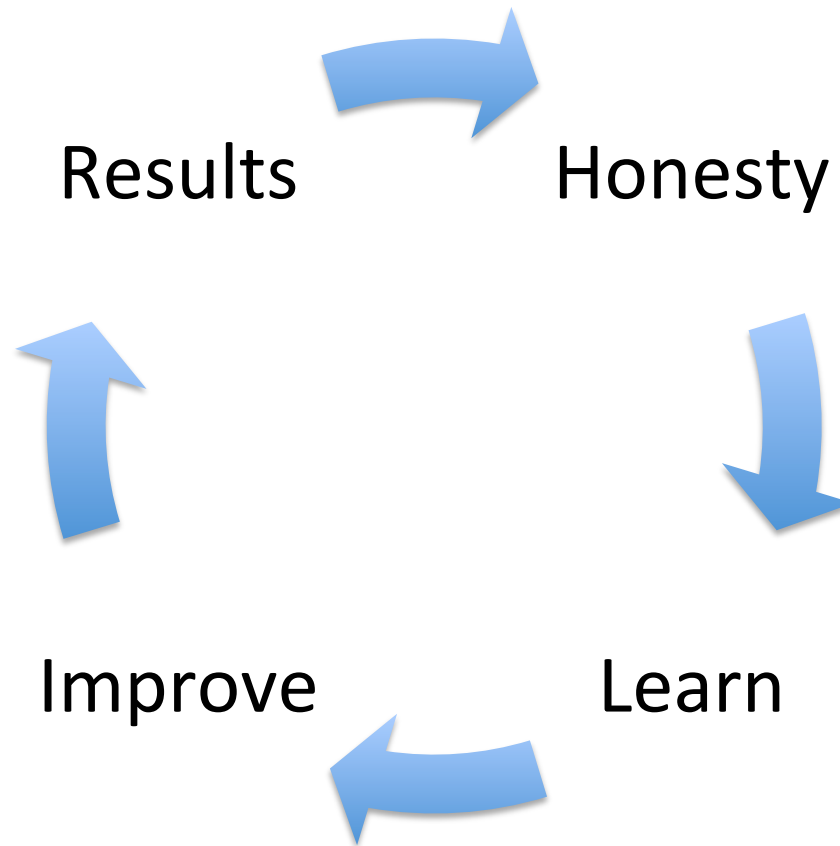
Google Blame-Free Post-Mortems

Action Items

- *How will we change process, technology, documentation, etc.*
- *How could we have automated the problems away?*
- *How could we have diagnosed more quickly*
- *How could we have restored service more quickly?*

Follow up (!)

Virtuous Cycle of Improvement

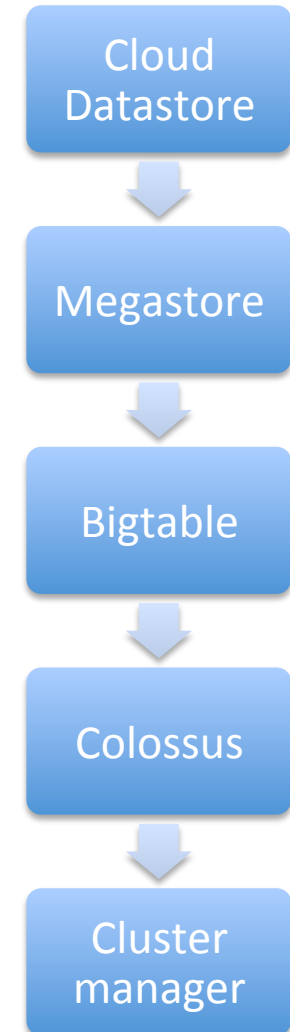


Organization: Service Teams

- **Small, focused teams**
 - *Single service or set of related services*
 - *Minimal, well-defined “interface”*
- **Clear “contract” between teams**
 - *Functionality*
 - *Service levels and performance*

Google Services

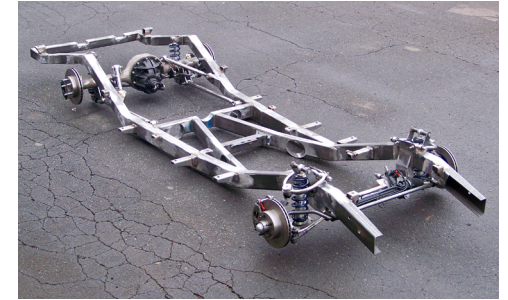
- All engineering groups organized into “services”
 - *Gmail, App Engine, Bigtable, etc.*
 - Self-sufficient and autonomous
 - Layered on one another
- ➔ Very small teams achieve great things



Organization: Ownership Culture

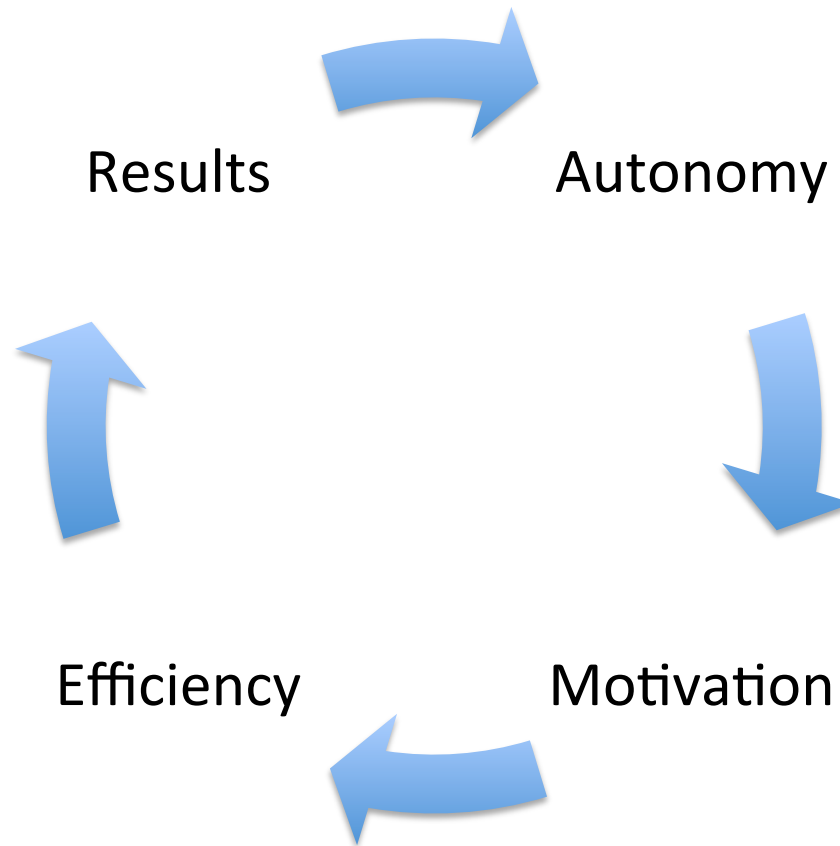
- Give teams autonomy
 - *Freedom to choose technology, methodology, working environment*
 - *Responsibility for the results of those choices*
- Hold them accountable for ***results***
 - *Give a team a goal, not a solution*
 - *Let team own the best way to achieve the goal*

KIXEYE Service Chassis



- **Goal: Produce a “chassis” for building scalable game services**
- **Minimal resources, minimal direction**
 - *3 people x 1 month*
 - *Consider building on open source projects*
- ➔ **Team exceeded expectations**
 - *Co-developed chassis, transport layer, service template, build pipeline, red-black deployment, etc.*
 - *Heavy use of Netflix open source projects*
 - *15 minutes from no code to running service in AWS (!)*
 - *Plan to open-source several parts of this work*

Virtuous Cycle of Ownership



Organization: Collaboration

- Act as one team across engineering, product, operations, etc.
- Solve problems instead of blaming and pointing fingers
- Leave politics to the politicians
- Bureaucratic games are not as fun as real-time strategy games 😊

Google Co-Location

Multiple Organizations

- *Engineering*
- *Product*
- *Operations*
- *Support*
- *Different reporting structures to different VPs*

Virtual Team with Single Goal

- *All work to make Google App Engine successful*
- *Coworkers are “Us”, not “Them”*
- *Never occurred to us that other organizations were not “our team”*

Why Are Organizations Slow?

People

Organizational Culture

Process

Process: Experimentation

***Engineer* successes**

- *Constant iteration*
- *Launch is only the first step*
- *A / B Testing needs to be a core competence*

Many small experiments sum to big wins

eBay Machine-Learned Ranking

Ranking function for search results

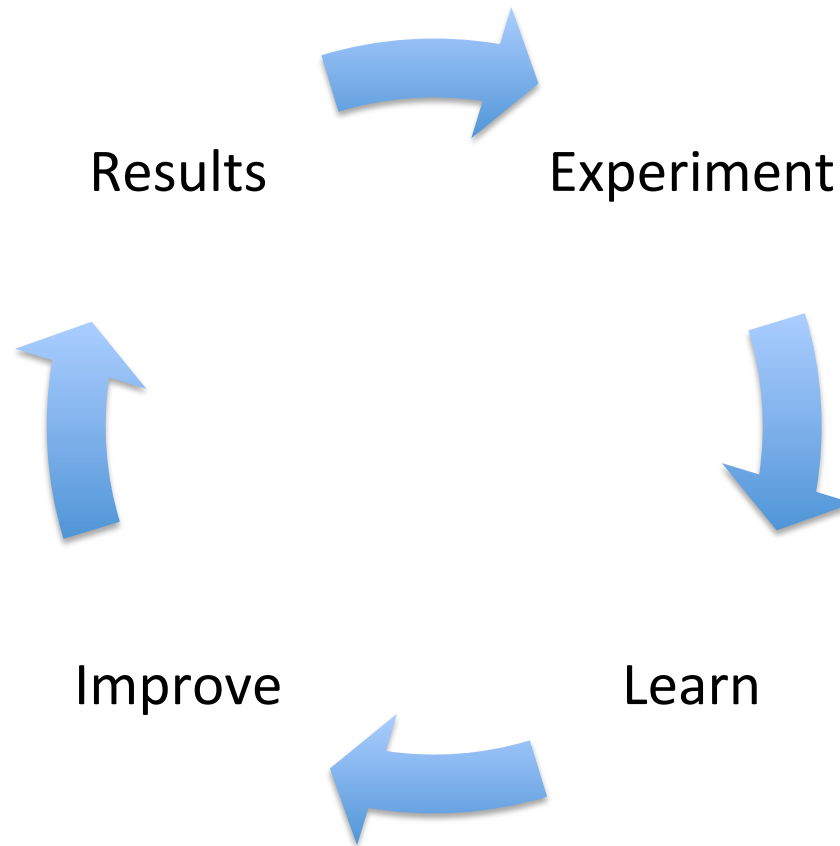
- *Which item should appear 1st, 10th, 100th, 1000th*
- *Before: Small number of hand-tuned factors*
- *Goal: Thousands of factors*

Experimentation Process

- *Predictive models: query->view, view->purchase, etc.*
- *Hundreds of parallel A/B tests*
- *Full year of steady, incremental improvements*

➔ 2% increase in eBay revenue (~\$120M)

Virtuous Cycle of Experimentation



Process: Quality Discipline

“Quality is a Priority-0 feature”

Automated Tests help you go faster

- *Tests have your back*
- *Confidence to break things, refactor mercilessly*
- *Catch bugs earlier, fail faster*

Faster to run on solid ground than on quicksand

Process: Institutionalize Quality

Development Practices

- *Code reviews*
- *Continuous Testing*
- *Continuous Integration*

Quality Automation

- *Automated testing frameworks*
- *Canary releases to production*

“Make it easy to do the right thing, and hard to do the wrong thing”

Google Engineering Discipline

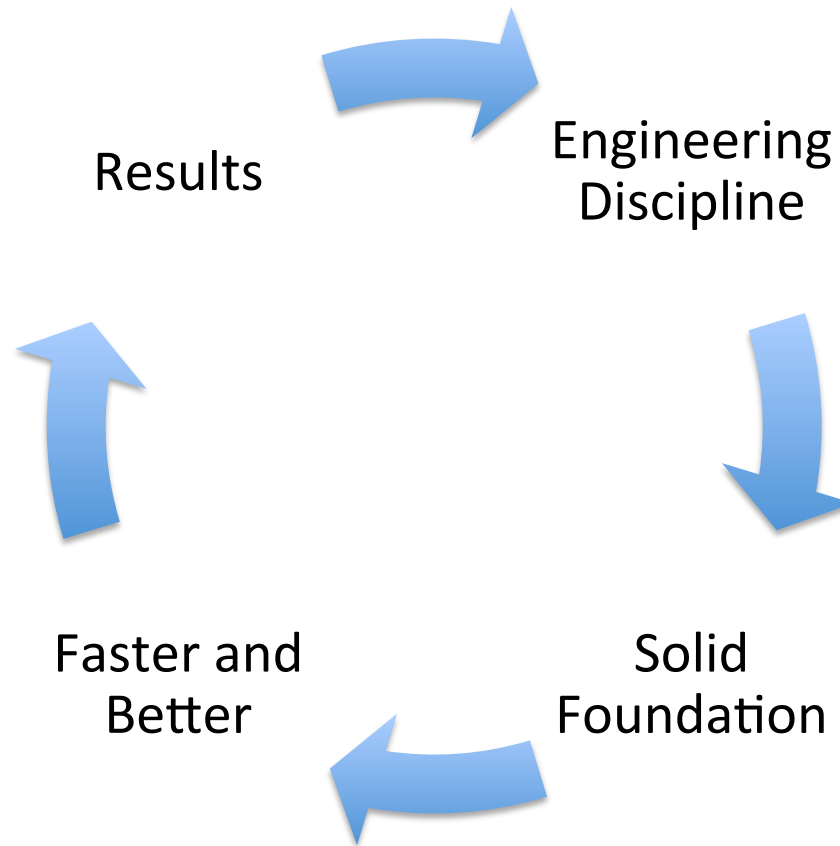
Solid Development Practices

- *Code reviews before submission*
- *Automated tests for everything*
- *Single logical source repository*

Result: Internal Open Source Model

- *Not “here is a bug report”*
- *Instead “here is the bug; here are the code changes; here is the test that verifies the changes” 😊*

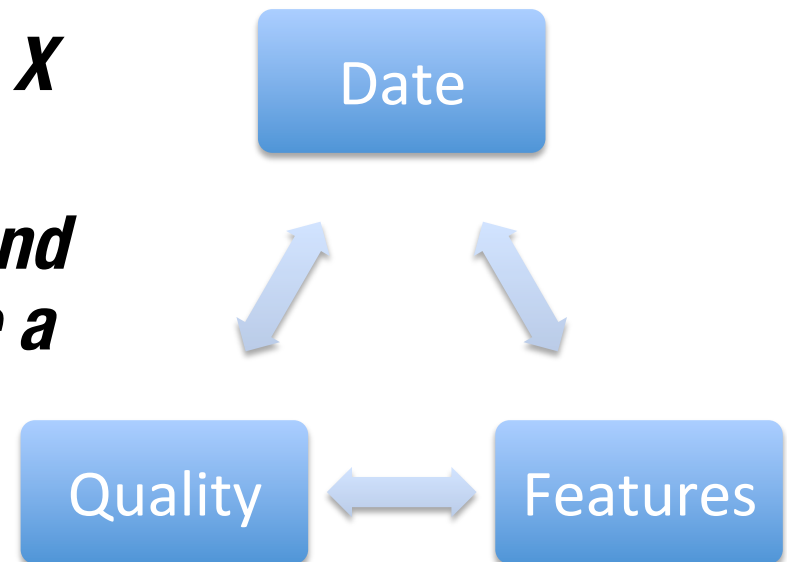
Virtuous Cycle of Quality



Process: Technical Tradeoffs

Make Tradeoffs Explicit

- *Every decision is a tradeoff: X or Y or Z*
- *When you choose features and a date, you implicitly choose a level of quality*



→ Be honest with yourself and your team when you are doing this (!)

Process: Technical Tradeoffs

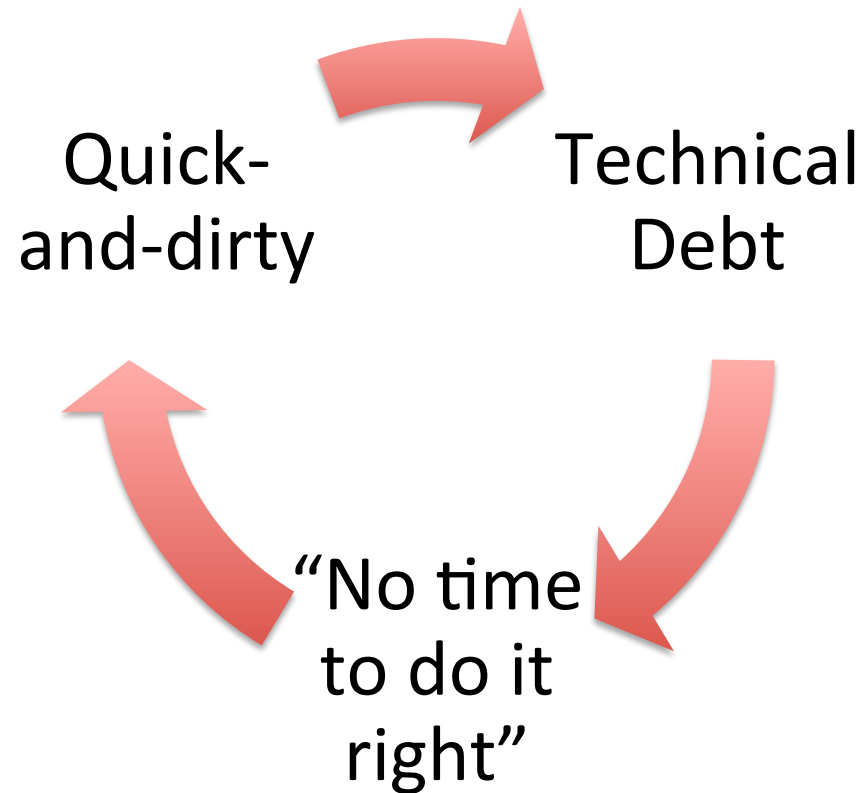
Manage Technical Debt

- *Plan for how and when you will pay it off*
- *Maintain sustainable and well-understood level of debt*

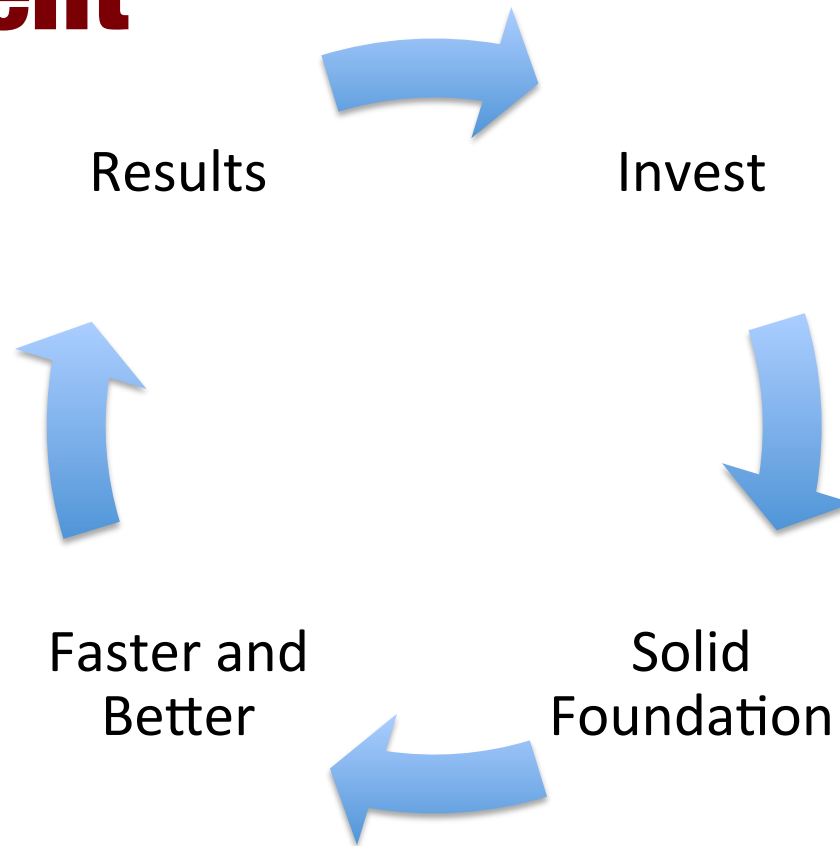
“Don’t have time to do it right” ?

- *WRONG – Don’t have time to do it twice (!)*

Vicious Cycle of Technical Debt



Virtuous Cycle of Technical Investment



Recap: How Can We Make Organizations Fast?

People

Organizational Culture

Process

Come Join Us!

KIXEYE is hiring in

SF, Seattle, Victoria, Brisbane, Amsterdam

rshoup@kixeye.com

@randyshoup

[linkedin.com/in/randyshoup](https://www.linkedin.com/in/randyshoup)

[slideshare.net/randyshoup](https://www.slideshare.net/randyshoup)

